



Actualizing IT Solutions

Experienced Validation

Draft last updated by **Broennimann, Gerry** on **07 Oct 2013**

100%
Complete

Overview

Required fields are marked with an asterisk (*) and must be completed for submission.

Actualizing IT solutions involves implementing all activities that transform information technology from a vision to an actual working solution and continuing to work with the client to ensure continued client satisfaction with IBM. It involves defining a solution to fulfill the client's needs, ensuring the technical viability of the solution within the client's environment, removing technical inhibitors to client acceptance of the solution or implementing that solution to the client's satisfaction. Actualizing may involve using demonstrations, proofs of concept, benchmarks, custom application development, package solution implementation, or integration services to deliver outstanding client value.

Attaining the Experienced level will mean you are considered Level One IT Specialist Certified.

Please supply all required content in your own words and be concise when providing information.

Introduction and guidance

The capability level validation package has been designed to document the attainment of newly-demonstrated Career Framework capability levels. It does not include assessment of performance on a project.

The validation process has defined who can access the document -- the employee and reviewer(s). Each individual has a different role and therefore may not have access at all times during the process. IBM confidential information should not be included in your validation package.

The phases of the review process, what occurs during each phase, and who has access to the document during each phase are described in the Help material. Once you have completed the package, click the 'Submit' button at the bottom of the page.

Specializations

The following specializations are selected for this package. You may change these selections, but this will change the content of this package, particularly on the "Evidence" tab.

Specializations:* Application Development and Maintenance
Support

Current level for Actualizing IT Solutions

Level	Specialization (if applicable)	Status
★★★★ Foundation	Support, Application Development and Maintenance	Transition: 17 Aug 2011

Employee information

Name: [Broennimann, Gerry](#)
Preferred name:* Broennimann, Gerry
Intranet address: gerry.broennimann@ch.ibm.com
External number: 41-58 333 33 74
Tie line number: 3-3374
Business address: Vulkanstrasse 106, Zürich 8010
Date joined IBM: 01 Apr 2007
Originating organization: Z1
Originating division/departm ent: /513315

Manager information

Name: [Musio, Ilario](#)
Intranet address: ilario.musio@ch.ibm.com
External number: 41-58 333 35 60
Tie line number: 3-3560
Business address: Münchenstein 4142

Manager Reviewer

The default manager reviewer is your Blue Pages Manager. If your Blue Pages manager does not provide you with business direction you should select your Functional Manager as the Alternate Reviewer. If selecting anyone other than your Functional Manager, you must first discuss with your Blue Pages Manager; your Blue Pages Manager will be notified upon package submission if an alternative manager reviewer is specified.

Manager Reviewer: First-line manager

Target submission date

You may optionally want to specify a target submission date, which you may want to include on your development plan.

Target submission date:

Mentor

As it relates to this process, a mentor is someone you designate to help you with your validation package. At a minimum, this person should hold the same capability at the same or higher level for which you are applying. Click the 'Share Package' link above to allow your package to have read only access to your mentor or anyone else to get feedback before you submit this package.

If you need help finding a validation package mentor, please contact the Capability Center for suggestions or go to the Explore Mentoring web site [see help link above for more info] for general information about finding mentors.

Mentor: Stefan Schenk

Intranet address: Stefan.Schenk@ch.ibm.com

Experience

Document how you meet the experience requirements for this capability.

Required experience

You must include at least two (2) experience profiles.(1 within last 3 years, 1 within last 5 years)

Project name	Client name	Status	Last modified	Actions
VISTA PV	IGS GmbH	Completed	06 Oct 2013	View profile...
VISTA IK	IGS GmbH	Completed	06 Oct 2013	View profile...

Experience profile

Engagement overview

Engagement name:*

VISTA PV

Client name:*

IGS GmbH

Nature of engagement:*

Tester / Architect

Location of engagement:*

Zurich, Switzerland

Name of your employer:*

IBM Switzerland

Total engagement duration:*

From 01 Jan 2006 To 31 Dec 2012

Your involvement:*

From 10 Mar 2008

To 31 Dec 2012

Business opportunity or problem

Your description should address the following questions:

1. Describe the nature of the engagement in terms of the business problem it addressed/solved and how it related to the client/organization's business and mission.
2. Define the application and the expected benefits to the client from its successful implementation.
3. Describe the scope and complexity of the problem in terms of phases of development or in terms of size (number of processes, client environment, number of modules, etc.).
4. Describe your relationship and communications with client management.

Answer:*

In 16 of 26 Swiss States (Cantons) & Principality of Liechtenstein (LI) the public social security state organizations (SVA or AK) have formed a jointly held IT Company IGS GmbH based in St. Gallen, which is the IBM Customer. IBM Switzerland together with IBM Remote Delivery Centre (RDC) in India have built new business applications for these 16 Swiss State Social Security Agencies (SVA) and the Principality of Liechtenstein based on JAVA, DB2, Websphere and Eclipse Rich Client.

VISTA is an engagement which was initially designed to replace the entire legacy system (mostly Host based) of the supported SVAs. The new system should be easier to handle for the users, tighter integrated in the workflow and among the business applications, more stable and easier to maintain and configure. Also, the yearly costs for license fees should get reduced drastically. VISTA was developed by IBM Switzerland, supported by teams from IBM Germany and the Remote Delivery Centre (RDC) of IBM India.

After some delivery issues and an increasing delay, the VISTA project has been divided into several phases (E1 - E3). Phase E1 aimed the rollout of the pension cluster applications (RE, HE, EL, IK). Later on, this phase was further divided into E1a and E1b. In E1a mainly the partner administration ("Partner Verwaltung", PV) should be delivered, along with the Back Interface (BI) to replicate VISTA data to the legacy systems.

In this profile - later on referenced as [VISTA PV] - the module PV should be described in detail. PV is a cross application which provides interfaces and basic insurant data for the business applications, as well as for external and legacy systems. With the new PV, the users got a well-arranged user interface with object tree, menus, dialogs and tabs. The usage of a role-based model enhanced the business possibilities even further.

For PV, a total of 92 GUIs were defined and implemented. Each of its fields follows one or more business rules, thus minimizing invalid data on the production environment. 276 Interfaces for general usage or customized for an application have been defined within PV. Seven WorkPerformers (chains of process steps) assure a precise, step-by-step work execution out of ELAR Advanced (external workflow management system). 695 business rules exist in PV which refine the 73 use cases. Six batches mainly steer the replication of PV data to other systems.

My roles, tasks and interaction with the client are described in more details in the solution section below.

(2513 of 3000 characters)

Solution

Briefly describe the solution you implemented, deliverables prepared, technology used, etc. Your description should address the following questions:

1. Describe your role in the engagement and your specific tasks, responsibilities, and accomplishments, including your role in planning the effort, tracking progress and reporting to the IBM and Client project management. You should be able to show how you performed as a Technical Lead IT Specialist in the development, implementation and/or management of the major project or subsystem.
2. Summarize the key technical decisions you made, the reasons for the decisions, and the alternatives that were considered.
3. Describe the major problems or obstacles your encountered, and the actions you took to overcome them.

Answer:*

As IT Specialist at IBM Switzerland, I had an architectural and test leading role in the module PV and started this engagement in March 2008. Initially, I was mainly responsible for the testing part, addressing the clients quality requirements. More than 600 testcases, created and maintained in ClearQuest Test Manager (CQTM) and Rational Manual Tester (RMT), had to be re-evaluated and retested by IBM, IGS and customer testers from the SVAs. In a later phase, the most important of these testcases had to be automated in a sub-project; the progress on it was closely monitored and evaluated by the customer's management. As Application Test Team Lead (ATTL) I tracked and led the progress in both phases and was responsible for a successful automation rate of the PV testcases during a 3-weeks workshop in India.

After the lead architect in PV left the project I took over his responsibilities and continued a close collaboration with the IGS product manager. He addressed new business requirements in change requests, for which I made the technical specification and the handover to RDC for development.

Later on, I additionally took over the role as application lead for PV. It was during a project phase where a tight time and budget plan had to be followed. Using several planning tools I was able to balance the workload among the team, schedule defects and changes for sub-releases and to report the development progress to both IBM and IGS management. During this engagement I also made code changes on the Back Interface (BI) module and analyzed replication problems.

Key technical decisions in my responsibility were the implementation of the notification concept, integration of an external application (GILAI) and complex data cleanup activities. I have simplified the new notification concept significantly, as the old implementation was too complex to handle and maintain. I interviewed the business application lead architects and product managers to evaluate what is really required and what is obsolete. For GILAI Integration I had to decide between two replication concepts. I could prove that one of these concepts is already in place and could be extended with less effort.

The data cleanup in PV was an ongoing challenge. The introduction of the new social security number (NSS) in Switzerland, business rule changes, migration errors and mass updates (partner reconciliation from tax data) led to service requests with sensitive mass corrections, as impact on other applications and systems had to be considered. Using careful test runs and clear instructions for the script execution these risks could be addressed sufficiently.

(2653 of 3000 characters)

Results

Assess the overall success or failure of the project from the standpoint of:

1. Client Satisfaction
2. Quality of deliverables
3. Performance of the IBM team

Answer:*

A clear indication of the client satisfaction was the collaboration with the PV product manager, my counterpart on customer side. Over time we achieved to enable and maintain a mutual trust and we could enjoy many fruitful discussions with clear decisions as outcome. PV was the first application we could set live and it could remain its stability. Some concerns were raised regarding the capacity: as the team in LDC was reduced over time to me only and I got increased responsibilities for another application, we had to postpone some activities due to restricted bandwidth. A clear prioritization and the settlement of fix dates could address this issue sufficiently.

During the Delta Development (DD) phase of the project the long release cycles were used for intensive test activities and quality improvements of changes. The limited test data in RDC could not always address all NFRs, even though they were not specified by the customer. But test results and experiences from the System Integration Tests (SIT) with production data could be integrated in follow-up development activities. Examples are replication batches, which had to handle a huge amount of data. A workload splitting logic resolved the initial memory and timeout problems.

For a long time, PV had very talented developers and testers in RDC which were able to judge the requirements and to think out of the box. In this team constellations, the most critical changes could be implemented and delivered without severe defects and on time. The team performed very well in RDC and also the landed resources (developers and testers) have shown a passion for the project. Conversely the test automation workshop in India overachieved the targets by far.

Some negative influences to all mentioned points (client satisfaction, quality of deliverables and the performance of the IBM team) had to be addressed when key resources were out-phased in later project phases. It has to be mentioned however that without landed resources and no access to SIT environments problem reproduction was difficult, sometimes even impossible.

(2092 of 3000 characters)

Lessons learned

Your description should address the following questions:

1. What might you have done differently on this project?
2. What lessons did you learn from this experience?
3. How did this change your behavior or decisions on subsequent engagements? Consider the entire solution lifecycle from strategy, design, implementation and management through to completion.

Answer:*

Within most of the VISTA applications there is a rather strict separation of roles (project managers, architects, developers, testers etc.). Architects with close customer contact are in LDC (Local Delivery Centre, Switzerland) only, whereas the developers and testers reside in RDC India. This restricted sometimes a quick problem analysis: RDC could not reproduce a problem as they did not have such data constellations as they exist in production. The other way round, the architects didn't have a working debugging environment as the setup for it is very complex and not forced in LDC. If I could start again, I would push the setup of a workspace which would allow debugging on SIT environments. One defect was open for over

a year as RDC was not able to reproduce the problem. A debugging session with a landed resource (even though he was from another application) identified the problem within minutes.

Even though the work in Global Delivery setup is very interesting, it requires a clear communication and also an understanding of its constraints respectively the requirements from RDC. Missing or different testdata can be a reason for non-reproducible results, but it's not always an excuse for rejecting defects without proper analysis. I learned that it's a huge benefit knowing India and the colleagues there personally. Nevertheless, you can't always be just the nice guy. Having the responsibility over a module sometimes requires a certain pressure on your colleagues. Showing (formal) appreciation for a good result is very appreciated, though.

I developed a strong out of the box thinking. Requirements in changes or defects have to be verified critically. If a requirement is unprecise or effects on other applications / systems are not considered in the beginning, a changed module can cause more problems than it solved. This is especially important in the design phase. For the implementation and testing phase a spot check via remote desktop connection ensures that LDC can test scenarios which might not be obvious to RDC developers. It is however no guarantee for a proper delivery and does not reduce the required test activities. From the management point, even if defect and change tracking might get awkward, depending on the team, it can get necessary and prevent delivery delays. Alternatively or additionally regular meetings / calls support an ongoing overview over the open tasks.

(2414 of 3000 characters)

Experience profile

Engagement overview

Engagement name: *	VISTA IK	
Client name: *	IGS GmbH	
Nature of engagement: *	Public Sector	
Location of engagement: *	Zurich, Switzerland	
Name of your employer: *	IBM Switzerland	
Total engagement duration: *	From 01 Jan 2005	To 30 Sep 2013
Your involvement: *	From 01 Sep 2010	To 30 Sep 2013

Business opportunity or problem

Your description should address the following questions:

1. Describe the nature of the engagement in terms of the business problem it addressed/solved and how it related to the client/organization's business and mission.
2. Define the application and the expected benefits to the client from its successful implementation.
3. Describe the scope and complexity of the problem in terms of phases of development or in terms of size (number of processes, client environment, number of modules, etc.).
4. Describe your relationship and communications with client management.

Answer:*

The general VISTA setup is already defined in the PV experience profile, therefore I don't repeat this part here but focus directly on the IK application.

IK stands for "Individuelles Konto" (individual account) and is the basis for calculating a pension of an insured person. All earned income, duration of contribution and bonuses for care-taking are entered into the individual account. They serve as a basis for calculating an old-age, survivors or invalidity pension. Anyone who would like to know if the contribution period is complete or whether an employer in fact paid the contributions deducted from salaries, can send in a written application for a statement of the account to the compensation fund they most recently contributed.

The new implementation should be easier to use with modern GUIs and should be more integrated with the entire system of the SVAs. New and reworked batches should further improve the automatic processing of data reconciliations with other business applications and the central compensation fund (Zentrale Ausgleichsstelle, ZAS) which would lead to tremendous cost savings for all SVAs using VISTA.

For IK, 221 detail use cases in 17 groups had been identified and designed. The batch plan contains 47 different batches for IK. 32 data extractors are responsible for creating correct printable documents (letters, lists and ELAR postbasket entries). 43 ELAR work performers assure a consistent execution of process steps, integrated in the ELAR workflow. 35 GUIs and a comprehensive tree view provide overview and secure data operation for a business case.

One of the main challenges for IK was it's role as a cross application for the pension system and other SVAs through ZAS. As the IK application of an SVA requires to send, receive and process data from all other SVAs, any delayed or unprocessed requests potentially exposes one SVA to the entire social insurance system in Switzerland (see CC7).

I took over several roles and responsibilities within the IK module and therefore also different relationships with the customer. They are described in detail in the solution section below.

(2127 of 3000 characters)

Solution

Briefly describe the solution you implemented, deliverables prepared, technology used, etc. Your description should address the following questions:

1. Describe your role in the engagement and your specific tasks, responsibilities, and accomplishments, including your role in planning the effort, tracking progress and reporting to the IBM and Client project management. You should be able to show how you performed as a Technical Lead IT Specialist in the development, implementation and/or management of the major project or subsystem.
2. Summarize the key technical decisions you made, the reasons for the decisions, and the alternatives that were considered.
3. Describe the major problems or obstacles your encountered, and the actions you took to overcome them.

Answer:*

I joined VISTA IK in 2010 in an architectural role just before phase E1a went live in the pilot SVA, initially mainly for assisting in the AMS organization: the other architects at that time had a good knowledge about the application itself but no experience in maintenance processes and incident management of a productive environment. As I already supported VISTA AMS in phase E1a I could instruct the team on the processes, tools and work products to be used for analyzing and resolving tickets. Obviously I also worked on incidents, problems and requests for service (RfS) myself. When the ticket backlog was increasing more and more, I represented IBM in a commonly formed "IK Taskforce", consisting of representatives from the SVAs, IGS product management, IGS service manager, IGS incident management and IBM architects. In weekly calls I had to update the participants about urgent problem tickets and action plans to reduce the backlog.

In 2012 I assisted the application leader in managing the capacities of the team members in Switzerland and RDC India and planned the development of changes and defects. I updated capacity sheets and tracked the development progress using ClearQuest and reports from the project office. It was a close collaboration with the Product Manager from the customer as we reported the project status weekly to both IBM and IGS management in a short presentation and status sheet.

Still, my primary role was a technical one. In this I worked on dozens of Changes, covering all phases of development: estimation, specification, implementation, verification and roll-out. While some were relatively small, others had a huge impact on the business. A key technical challenge was offline massbooking, a batch which had to deal with up to 100'000 bookings to be processed and complex lists to be printed. In the initial concept no NFAs had been specified and tests with former test data passed at that time, but for bigger mandators having several times this workload, we encountered more and more batch aborts in production. Instead of a try-and-error attempt to a selective optimization I encouraged the customer to assign a change to IBM for an entire redesign of the batch (and separate some activities to another new batch).

Due to it's data driven nature, IK soon had to process many data cleanup activities. While many issues could be covered with SQL scripts, some data required to consider complex dependencies. I set the basics for a new clean-up batch which could re-use existing services to create or update IK data, compliantly with all business rules.

(2595 of 3000 characters)

Results

Assess the overall success or failure of the project from the standpoint of:

1. Client Satisfaction
2. Quality of deliverables
3. Performance of the IBM team

Answer:*

The IK application received mixed feedback statements. Both IBM and IGS had great interest in a successful go-live with the pilot mandator and meeting all criteria for follow-up roll-outs. It is a known fact (though one does not speak it out loud) that IK had actually not been ready for being set live: many business scenarios had not been covered in the test phases and were discovered in production only.

As IK is exposed to ZAS and consequently to other SVAs, unprocessed requests immediately pointed to the new VISTA solution and created pressure on IGS and consequently IBM. Several escalations from the SVAs impacted the client satisfaction negatively. This satisfaction declined when several batches could not cope with the increasing workload anymore and crashed repetitively. Still, I could also point to the fact about the sheer amount of business cases which are processed successfully in the online and offline business.

A mixed picture regarding the quality of deliverables can be taken, too. The basic application and the core functionality offered a huge benefit for the SVAs and reduced handling errors and manual effort significantly. Nevertheless, the IK team also delivered changes with many findings and defect fixes which corrupted previously working components and led to data correction and urgent patches (iFixes). Such events merely occurred in the AMS phase when experienced developers left the project and less experienced resources took over.

The IBM architects in IK most notably received positive feedback from the customer and the management. We could establish a productive collaboration with IGS and the end users from the SVAs. We had to deal however with limited capacity for all tasks and sometimes with non-performing developers of the RDC team. The experienced and well-performing developer had to compensate the lack of skills of their colleagues, which wasn't always enough and caused changes in the delivery plans.

(1955 of 3000 characters)

Lessons learned

Your description should address the following questions:

1. What might you have done differently on this project?
2. What lessons did you learn from this experience?
3. How did this change your behavior or decisions on subsequent engagements? Consider the entire solution lifecycle from strategy, design, implementation and management through to completion.

Answer:*

Even though I didn't have much influence on the project setup, especially during the development phases before the AMS setup, I identified some critical points which I would have done differently.

The global delivery (GD) setup, combined with urgent incidents from production, led to a heavy dependency on quick analysis work from RDC. The RDC team however is only tracked by defects and changes and not on LDC analysis support. Such support activities were "unproductive" or even "lost" time for RDC, at least according to certain managers. Consequently, support activities for LDC had low priority for them. In a more service-oriented setup AMS RDC should have the instructions and guidelines to treat supporting tasks with the same (or even higher) priority.

No IK architect in LDC ever had a workspace in RSA (Rational Software Architect) which could have been used for debugging. And I encountered many situations where the root cause for a problem could have been found very easily by just re-executing a step in debug mode with data copy of production. Describing the problem and provide appropriate test data

to RDC to (hopefully!) reproduce an issue was a huge overhead and in many cases not even successful.

Especially for a data-driven application like IK I see many points to improve in the GD setup. Either there is always a landed resource (an RDC developer) in LCD with the capability and capacity to support the analysis on code level. Otherwise RDC should have access to the same (test) environments like LDC. If this isn't possible due to legal reasons like it is in VISTA, a proper data anonymization must be in place, so that at least most of the failing scenarios could be reproduced in RDC.

Finally, an accurate test coverage with working and an up-to-date regression suite would have prevented some critical side-effect defects. Addressing lacks in the test coverage area and identifying measurements for improvement would be on high priority in a next engagement.

(1983 of 3000 characters)

Evidence

Document a minimum of two (2) examples for each of the capability statements.

You may also embed images, diagrams, and tables, but should resize these if necessary to ensure they are printed correctly.

Core capability statements	Examples required	Status	Action
Written Communication (CC1)	2 required	2 completed 2 attachments (880 KB)	View evidence...
Technical Direction (CC3)	2 required	2 completed	View evidence...
Negotiation (CC4)	2 required	2 completed	View evidence...
IT Project Plan Management (CC5, CC9)	2 required	2 completed	View evidence...
Team Leadership (CC6)	2 required	2 completed	View evidence...
Business Aspects (CC7)	2 required	2 completed	View evidence...
Solution Input to Winning Bids (CC8)	2 required	2 completed	View evidence...
IBM Business (CC11)	2 required	2 completed	View evidence...
Solution Development (CC12)	2 required	2 completed	View evidence...
Problem Resolution and Analysis (CC13)	2 required	2 completed	View evidence...
Personal Impact (CC14)	2 required	2 completed 1 attachment (33 KB)	View evidence...

Core capability statements	Examples Status required	Action
Interface to Architecture (CC15)	2 required 2 completed	View evidence...
Asset Re-use and Harvesting (CC16)	2 required 2 completed	View evidence...
Verbal Communication (CC2)	2 required 3 completed 3 attachments (3.2 MB)	View evidence...

Written Communication (CC1)

Created and delivered compelling written communications in an organized and persuasive manner, adapted style for different audiences in project deliverables and in a client environment.

*** List two documents that were published or provided to internal or external clients and that demonstrate your ability to effectively communicate decisions and designs in your technical specialty. For each document, provide the name of each document and a short description of the purpose of the document.

Guidance for this capability statement

Your examples should show how you communicate your technical expertise in written form. Keep in mind that your validation package is the main document that demonstrates your written communication ability. Try to be specific in both the actions and results.

Example 1: VISTA PV - Social Security Number Reconciliation

Duration:* From 01 Feb 2013 To 30 Jun 2013

Project or major activity:* VISTA PV - Social Security Number Reconciliation

Description:*

Filename:
44085_NNSS ergänzen aus UPI_Konzept_v1.0_compressed.ppt

Purpose:
In Switzerland, a new Social Security Number (NNSS in German) was introduced to uniquely identify each insurant person. As the databases in all VISTA-Cantons initially only contained the old number and an NNSS could not be assigned to each case, many partners in VISTA could not be identified correctly and duplicate partners were created.

The customer asked for a solution to reconcile the VISTA data with the database of the central insurance agency (UPI-Registry). As initial tests have proven that normal SQL scripts would run into timeouts due to performance reasons and the amount of data, I created the mentioned concept which uses an iterative, status-driven approach which would allow an execution over several transactions.

The customer appreciated this concept for the traceability and flexibility of this approach.

(899 of 5000 characters)

[44085_NNSS ergänzen aus UPI_Konzept_v1.0_compressed](#) (PPT, 1.1 MB, 13 Aug 2013)

Example 2: VISTA PV - Handover document for new Architects**Duration:*** From 01 Jan 2012 To 01 Jul 2012**Project or major activity:*** VISTA PV - Handover document for new Architects**Description:***Filename:
TOD999_PV-Overview + Handover.doc

Purpose:

This document serves as a handover document for new PV architects. It explains some complex and important topics in a summarized form, addresses open issues and workarounds and provides links to important documents and tools. With this document, a new PV architect should be able to get a complete overview of all relevant tasks in PV.

(385 of 5000 characters)

[TOD999_PV-Overview + Handover](#) (DOC, 494 KB, 13 Aug 2013)

Technical Direction (CC3)

Planned the work, formed a team to perform the work and guided the team to complete a scope of work.

Provide two instances where you set the technical direction and constraints of a project or engagement and monitored compliance. Only provide instances where your role in the effort was the lead for your technical specialty in the project in the project (or a significant subsystem or component.).

Guidance for this capability statement

Here, you want to include examples where you provided the vision for the technical solution. How did you ensure that the team understood and was conforming to your vision? Use examples that show you completed a scope of work through the use of a team of individuals. Show that you bring in the right resources to move the opportunity or project along with a focus on the technical direction that should be taken by the client. Show how you organized the tasks in your assigned scope of work into manageable work items, understood all the dependencies related to the work and worked with a team to complete the assigned work by the due date.

Example 1: VISTA PV - Subproject Integration of "GILAI" into VISTA**Duration:*** From 06 Jul 2012 To 30 Jun 2013**Project or major activity:*** VISTA PV - Subproject Integration of "GILAI" into VISTA**Description:***

For the "Invalidity Assurance" (Invalidenversicherung, IV) business, the customer decided to buy an existing solution ("GILAI") from another provider (GLOBAZ) and have it integrated into the VISTA context. At the beginning it was only clear that PV data "somehow" needs to be replicated to GILAI. That was the point where the customer came to me. First, I drafted a concept which required a decision about the replication mechanism. Central Architecture, me

and the customer agreed on my suggestion to use a very similar way like which was already in place to replicate to the billing system (Abrechnungssystem, AS). The other possibility to replicate data in the way the Back Interface (BI) does for the legacy systems would have been more complex and RDC had no know-how about it because BI was developed in LDC only.

After this decision, I defined milestones for this subproject, considering the target release date and the team's capacity. Then, I invited architects from GLOBAZ, SVA users who knew the IV business and the product manager from IGS to Zurich in order to have a common understanding of the solution and to finalize a mapping table. GILAI has a very different data structure than VISTA, but on attribute level we were able to map each required attribute from GILAI to a corresponding field in VISTA PV. In this meeting we could also reduce the complexity by reducing the required mapping tables.

In another meeting with GLOBAZ and IGS, we could agree on the data access on VISTA, also considering data security by using a federated database system. After this clarifications I could further refine the planning and define all "features" (subtasks for different development teams):

- PV feature to implement the replication batch
- Another PV feature for the possibility to have an instant replication via button click
- DB feature to create new tables and views
- CS (Common Services) feature to extend an existing clean-up batch

Each feature contained effort for analysis, implementation and design and was planned according to available capacity for each team and their cross-dependencies. As the replication batch was dependent on new constants (Code Management values) and the new tables the DB team had to start with it's feature. The next subrelease was used to implement the replication batch. With this subrelease, the customer could already start with some testing activities, whereas the instant replication button and the clean-up batch had lower priorities.

(2472 of 5000 characters)

Example 2: VISTA IK - Redesign Batches

Duration:* From 01 Mar 2013 To 30 Sep 2013

Project or major activity:* VISTA IK - Redesign Batches

Description:*

In IK, most of the batches have been designed with priority on business correctness and not on performance. Also, for a long project phase the customer tested with a "medium-size" mandator only (Canton St. Gallen SG) where many test cases have been executed successfully with this workload. But as the biggest mandator (Canton Zurich ZH) went live, we experienced performance issues like batch time-outs and rollbacks.

After many incidents in production and negative feedback from mandator ZH, IGS accepted my suggestion to have changes to improve the performance for the batches with the biggest workload. I have identified several batches and suggested a priority, assigning the mass booking batch as top priority. I created the following Change Features:

- IK Feature to redesign the mass booking batch (modelling, coding)
- BC (Batch Controller) Feature to update the batch plan (new batch definition)
- OS (Output System) Feature to update the error list document

I planned the development and target releases according to the timeline when IK would have a local landed resource from RDC. I could convince the customer about this dependencies and target releases, as I also mentioned the possibility to have the possibility to sit together

with the developer and directly address inputs, questions and corrective measurements. Plus, we could directly develop against production data on SIT (System Integration Testing) environments.

Together with another architect and the developer I identified the biggest bottlenecks and defined corrective measurements, e.g. loading only IDs during the "calculate workload" process instead of entire Data Entities (DEs). I gave an example of another batch from PV where we introduced a workload splitting mechanism which would prevent timeouts. And, referring to a current production issue, I suggested to exclude the list printing task into a separate new batch.

The "external" features (BC / OS) were delivered on time which also allowed a first version of the new batch in an early subrelease. As promised, the customer could test this first version with a huge data load and immediately address small correction measurements. The new batch was set live successfully and almost more noticeable, we were finally able to process a hanging mass booking portion and deliver the required documents to the customer.

Based on this positive outcome, the customer agreed to have more batches reworked in a similar way.

(2440 of 5000 characters)

Negotiation (CC4)

Negotiated to establish alliances and agreements that created value and resulted in equitable solutions for the parties involved. Developed a strategy to achieve resolution and facilitated discussions that created a stable outcome that benefited all parties. Escalated issues requiring higher authority for resolution.

Document two situations in which you successfully led negotiations involving multiple issues. Describe each situation including the preparatory work you completed, the parties involved, the opening positions of each party, the issues which were raised and negotiated, and the final agreement reached.

Guidance for this capability statement

Your examples should show you lead negotiations, from planning through resolution, resulting in a successful and stable outcome for all parties. Your examples should show you can manage multiple issues and objectives to a successful conclusion. The conclusion could be to walk away or to escalate.

Example 1: VISTA - Finding Clearing Board / Release Manager

Duration:* From 01 Jan 2011 To 30 Sep 2013

Project or major activity:* VISTA - Finding Clearing Board / Release Manager

Description:

Beside my normal job roles I am the delegate of the VISTA release manager on IBM side. Along with tracking the implementation state of defects and changes, the most noticeable task is executing the Finding Clearing Board (FCB), which is a telco twice per week.

In this meetings, the product managers and lead architects present new defects which were found recently and have to be planned for an upcoming release. In this calls the key is to identify the risks, severity, business impact, workarounds and efforts of each defect. All of

these facts have to be considered to suggest the final target release. For low and medium severe defects the target release is often quite easy to agree with. Critical bugs are a different story, however.

It happens that a defect is too severe to have it implemented with the next official release (which could be delivered up to three months later). In such cases, an emergency interim build (iFix) is required. As this is connected with high efforts on all sides (development, infrastructure, build team, service provider and test management), these iFixes need to be negotiated with all parties.

In this role I had to manage many iFixes. My responsibility was to get a confirmation from IBMs build and development teams about a delivery date and time. Sometimes the customer wanted a very early delivery which required overtime for several affected resources. Or the timeline was only reachable with weekend work which needed to be negotiated with the management in RDC India. On the other hand, I could agree in the name of IBM to deliver a very big package of defects but gaining one or two days of development and testing time.

There were also defects which had no visible impact on the end users at the SVAs, but generated many automated "event monitoring tickets" for IBM AMS. In such cases I also requested to deliver a defect earlier than planned (i.e. as iFix) in order to prevent floods of AMS tickets.

Despite the additional effort for iFixes, the client satisfaction is impacted very positively for each successful iFix, as we can prove to fix very important bugs in a short time.

(2125 of 5000 characters)

Example 2: VISTA IK - Backlog to normal service execution

Duration:* From 01 Mar 2013 To 30 Sep 2013

Project or major activity:* VISTA IK - Backlog to normal service execution

Description:*

VISTA was set live despite many open points, especially in IK. As IK is a data driven application sends and retrieves data from and to other SVAs, even small bugs can cause a massive amount of wrong data in production. Consequently, the IK ticket count for AMS was increasing since go-live.

Additional resources, an "IK-Taskforce" on Incident Management side, implemented changes and defects could stabilize the growing backlog trend. Also, at first the ticket backlog did not have any financial or contractual consequences. This changed however in March 2013, when it was decided to apply the service level agreements (SLA) for IK tickets, too. This required negotiations with several parties.

First, I could convince the IGS management that we can accept and handle ticket resolution times within the SLAs, but we would require a special treatment for the "old" tickets, otherwise dozens of tickets would escalate at once, with a heavy impact on the financial side. My convincing argument was that it would be of no benefit for the mandators (end users), as tickets can only escalate once, and then IBM had no further commitments on such a ticket and would concentrate not to escalate new tickets.

The commitment I had to give as IK application leader was to further negotiate with IGS product- and incident management about important and urgent tickets. We agreed on the following process: IGS would define small lists of tickets which should be prioritized from the backlog, i.e. moved into the normal operating mode including SLA resolving times. For each tickets in these lists I would (after the team's feedback) propose a date, when we could achieve a final resolution respectively on which date the normal SLA countdown would start.

For some tickets I could even arrange a quite late resolving date, as I could argue that the required effort to resolve a ticket would be much lower with a new release.

The outcome of all these negotiations are positive for all sides: tickets are resolved considering the real severity of the end users, and in a way and time frame to which both IBM and IGS have agreed. And so far, IBM didn't have to pay any financial penalties due to escalated tickets.

(2196 of 5000 characters)

IT Project Plan Management (CC5, CC9)

Identified those elements of a project plan that put the integrity of the specialty specific elements at risk and helped the client and or project manager by managing those elements so that the project was successfully completed. Demonstrated experience with creating a project plan, task tracking, capturing 'actuals', managing risk, managing teams, and management reporting.

Document two instances where you worked closely with the client or project manager to identify and address elements of the project plan specific to your specialty that put the technical integrity of the project plan or timeline at risk. Show how you mitigated the risk to both the technical or application aspects and the project milestones and tracked progress in the project plan.

Guidance for this capability statement

Your examples should show you are able to manage risks in a project plan specific to your area of expertise. You must be able to do this in routine situations, but may consult with mentors or require supervision for more complicated situations. . Examples can include planning an RFP response or a pilot using a logical approach (CVM, IBM Unified Method Framework, or Team Solution Design). Examples do not need to include project plan management tools such as MS Project or Rational Portfolio Manager.

Example 1: VISTA PV - Application Lead after Delta Development Project phase

Duration:* From 01 May 2011 To 30 Sep 2013

Project or major activity:*

VISTA PV - Application Lead after Delta Development Project phase

Description:*

After a project phase in VISTA (VISTA Delta Development), I additionally took over the role as application lead for the applications PV, IPV and BI in the new project phase "VISTA Change Project" and "VISTA Rollout". These were project phases with a rather strict budget and expectations from customer side. The tasks and challenges for my applications were:

- Identify and prioritize important changes
- Analyze and plan them according to the available budget and resources
- Request additional budget / resources in case of important new changes
- Track the overall progress plus the detail states of each changes
- Keep to the budget
- Include defects in the calculations / planning
- Report the status of the application to the management weekly

As all these applications were relatively small in respect of the resources, big changes got also a massive impact on the release planning and budget. The rollout of mandator

Liechtenstein (LI) had an additional risk: LI needed to support payments to accounts in foreign countries, something which was not foreseen for the Swiss Cantons. The technical restrictions resulted actually from the legacy system which was not able to support them.

This new requirement came very late from the side of mandator LI, but they announced that they cannot go live without this Change. The rollout deadline was fix, but I could successfully request an additional build before rollout. This allowed more testing time and one more iteration for fixing Defects. Technical problems with the legacy system were mitigated on an organization level: instead of a huge modification in the Back Interface (BI) logic, I analyzed which error messages respectively event tickets are to be expected and how Incident Management and the Legacy team could address them.

In the testing phase we found out that the data PV passes to the billing system (Abrechnungssystem, AS) were not compatible for certain countries. As I could prove that this is not a mistake on our side but a restriction coming from the SAP module, we offered to correct the cases in a new Change, to be delivered in an interim build (iFix resp. iChange) before go-live. This could be performed within our capacity, the budget had to be increased only slightly and the entire requirement could be delivered successfully on time.

(2294 of 5000 characters)

Example 2: VISTA PV - Test Automation

Duration:* From 01 Oct 2010 To 31 May 2011

Project or major activity:* VISTA PV - Test Automation

Description:*

By the end of 2010, IBM and IGS have agreed to initiate a test automation subproject within VISTA. The target was to create a suite with a huge list of automated AIT test cases which could be executed "on mouse-click" over night for release verification activities.

I have been identified to take over the lead for the PV test automation. The identification of test candidates and the organization of the preparation activities and handover to the test automation team were under my responsibility. In this time I reported to the VISTA test manager and my application leader.

In a team meeting we decided to establish a common tracking sheet for all applications. As all testcases were registered in ClearQuest Test Manager (CQTM) and could be flagged with different states, the Excel sheet was extended with the functionality to get the latest state of each testcase directly from CQTM.

After the capacity for PV had been defined I could start with the identification process. This was generally done by prioritization based on the testcase description. In this step a number of old testcases could also be marked as obsolete.

One of the first challenges we encountered was the fact that all AITs were written in German only. I definitely did not have the time to translate them to English for a handover to RDC. I addressed this risk by taking profit from a landed resource: in regular sessions, we went through some testcases; I explained in English what the testcase is about while she was taking notes. After each session, she could independently rewrite the testcases in English and assign them to the test automation team.

A second phase of this subproject was a workshop in India, in which testers from the SVAs were accompanied by IBM architects, in order to support and review the automation progress. The management set high progression targets which IBM should present to the customer. In order to achieve these targets, I identified groups of test cases which used the same set of dialogs, because "preparing" a dialog for automation was one of the most

time-consuming task. Before the workshop started I already communicated which dialogs should be prepared first. Like this, the PV team could reuse these components and progress very quickly. Finally, PV has overachieved the targets.

To ensure the quality, we made daily reviews of the testcases; for the test coverage I mapped the use cases with the test cases and defined a minimal coverage rate. Based on this result the remaining automation candidates have been assigned for further processing.

PV contributed considerably to the overall test automation target, which was a critical point for the acceptance criteria and an important requirement for the Go-Live of VISTA.

(2734 of 5000 characters)

Team Leadership (CC6)

Led a team.

Document two situations where either alone or with support or mentoring you led a team to perform a technical task.

Guidance for this capability statement

Your examples should show your experience leading teams in the development and implementation of technical solutions. You should reference projects where you were responsible for leading a technical team where your activities included contributing to the solution architecture and having responsibility for technical team deliverables. Teams can be made up of client, business partner or internal IBM team members. Whereas CC3 Technical Direction focuses on your vision of the technical solution, in this capability, this capability is focused on your team leadership - you may or may not have also have provided the technical direction.

Example 1: VISTA PV - Application Leader / Test Team Leader

Duration:* From 01 Apr 2008 To 31 Jul 2012

Project or major activity:* VISTA PV - Application Leader / Test Team Leader

Description:*

My first leading role in IBM was being application test team leader (ATTL) for PV. In this role I reported to the application leader and IBM's test manager. I supervised two testing resources in RDC and testers from the SVAs.

PV had a collection of around 500 application integration tests (AITs) which I had to maintain. The maintenance included to check if business rules have changed or any other development made test cases obsolete or candidates for an update. For this we ran a complete regression in a team with four customer testers and me as coordinator. I defined daily and weekly targets and assigned testcases across the team in a mutual understanding.

I let the team helping each other in a collaborative environment but tried to address their queries with priority. The progress was good and PV had an up-to-date regression suite as outcome.

A similar task was required when the automation of AITs started. I had only one tester from an SVA but also a landed resource from RDC and two test automation architects in India. Before a three-weeks workshop in India could start, we had to prepare many things, most obvious the translation of the German AITs. We could prepare enough AITs for the workshop

itself and we could therefore optimize the work distribution during the three weeks in India. PV has overachieved it's targets.

When the former application leader left the project I also inherited his duties and managed defects, changes, capacities and reports for the management for which I used ClearQuest, SharePoint and reporting templates from the project. I kept the established communication ways with the customer and RDC with weekly meetings and calls.

Despite some fundamental changes, PV remained to be a stable application and a team which delivered changes in a good quality.

(1797 of 5000 characters)

Example 2: VISTA IK - Application Leader

Duration:* From 01 Mar 2013 To 30 Sep 2013

Project or major activity:* VISTA IK - Application Leader

Description:*

For VISTA, 2012 was the year of the "accelerated rollouts": each month, a group of new mandators (SVAs) went live with the new system. During this time I was the deputy of the application leader. In his absence I managed the capacity planning, tracked the development progress and any risks for successful live-deployments.

With each new SVA in production, the required AMS effort for handling tickets and defects from production increased. In Q4/2012 the current application leader left the project and I took over, among my other roles as architect of IK and PV. During this time I put a high attention on the incident management. I established a biweekly short "standup meeting". As preparation, I went through the new tickets and added them in a tracking list. During the meeting, all team members provided inputs to these tickets. We marked the tickets as known or unknown problems and assigned them to the Architect who had the best background knowledge and who could resolve it with the least effort.

For known problems we listed the corresponding defects or changes. As this list was public and after some time everybody knew about open defects/changes, data corrections or at least which team member already worked on a similar problem and could provide help or take over a new incident.

Of course, the development tasks could not be ignored, as IK still had many changes and defects to implement. On one side, the RDC team had to be managed. I used weekly calls with the lead developer in RDC to ensure LDC and RDC plans were matching.

When IK encountered an increasing number of not-reproducible defects (on AIT environments in India) and changes with issues, I requested and initiated a landed resource for IK. The benefit for the entire team was enormous: problem scenarios from the SIT environments could be debugged and analyzed on the same systems. Changes could be discussed and handed over face-to-face and in a fraction of time. Additionally, the developer could provide valuable inputs for ticket analysis.

My role also expected a close contact with the customer, respectively the Product Manager of IK. We agreed to have a biweekly (short) meeting to ensure we both have the same understanding of the priorities. Additionally, we discussed and prepared the weekly status meeting for the management. We achieved to established an honest, productive communication between us and the spirit to face challenges as a team.

(2433 of 5000 characters)

Business Aspects (CC7)

Demonstrated understanding of the stakeholders' business needs and how they relate to your specialty.

Provide two instances that demonstrate you met this criterion. Examples must show how you made an explicit linkage between the technical specialty specific aspects and the business need.

Guidance for this capability statement

Your examples should show your experience determining the business problem the stakeholder is facing. Show how the stakeholder's business needs related to your technical specialty.

Example 1: VISTA PV - Configurable Integration of Changes

Duration:* From 01 Jan 2010 To 30 Sep 2013

Project or major activity:* VISTA PV - Configurable Integration of Changes

Description:*

As IBMers we continuously need to overachieve the client's expectations. Understanding his business is the key for a productive collaboration. Important is also to think out of the box and react to business requirements, which might be formulated via a change request in a very condensed form. In earlier phases of VISTA we had examples where changes needed to be reverted partially, as some dependencies had not been considered.

As a lesson learned, I targeted a flexible implementation for the bigger changes. Often this could be achieved via code management parameters. Even though PV does not have much business logic, as one of the most central application a wrong parameter could influence the entire system.

One example of configurable - and therefore future-proof - changes was the PV notification. In short, if something changes on the basic data of an insured person in PV, a message should be triggered to the business applications which could then handle such events automatically or creating a note for the users. In the conceptual phase, it was unknown which applications would implement this concept at all. And if, many details were open at that time. Considering these uncertainties the implementation had to be able to be compliant with future business decisions in this context. Therefore, the notification logic in PV addressed all undefined points with parameters. Second reason for it was the different organization of the SVAs. Whereas for some SVAs PV should inform the pension business when a person got married, it might be obsolete for other mandators. We delivered a solution which could handle entirely different usage profiles of the SVAs, from excluding one dialog or batch from triggering notifications to entirely turn off this mechanism - and be it only temporarily.

I used similar approaches for other changes with a big impact, like the integration of another system (GILAI) into VISTA or foreign payment addresses in Liechtenstein (see CC3 & CC12).

PV has also proactively suggested some changes, which we believed could be of great use for the customer. I understood for example the high usage of payment addresses - as paying out the insured persons is one of the core business of the SVAs. As the SVAs still use some legacy systems for some business applications, all payment relevant data has to be replicated from VISTA to legacy. As replicating bank basic data was not in the initial scope and manual database operations were required in case a bank changed or got created, I

could analyze the effort and impact to add this functionality. Unfortunately, the efforts were too high for the customer, so the feature had never been implemented, even though the costs would have been compensated already.

At least, we could help the customer to optimize the manual process and even outsourced part of it to RDC, enabling a time- and cost-efficient way for a core business process.

(2913 of 5000 characters)

Example 2: VISTA IK - Data Export to other mandators

Duration:* From 01 Mar 2011 To 30 Sep 2013

Project or major activity:* VISTA IK - Data Export to other mandators

Description:*

Understanding the IK business of the SVAs is as challenging as important. IK is an extremely data oriented application and business. But the structure, the quality and quantity of IK relevant data is just the beginning. It is rather the origin and target of the data that is sent from and to the IK application which is important to keep in mind: IK is a data receiver and provider from and to all other SVAs in Switzerland / Liechtenstein.

Simplified, the IK application of one SVA sends (among other) income data to the central compensation fund (Zentrale Ausgleichsstelle, ZAS) which by itself distributes the data to all other relevant SVAs. The other SVAs might or might not use VISTA. Delayed answers, unprocessed requests and data in a wrong format or wrong from a business point of view exposes the erroneous SVA to the entire social insurance system in Switzerland / Liechtenstein.

As lead architect it was my responsibility to assist the customer in setting priorities and to show him risks and chances in this core business. For example by suggesting optimization changes for the data export batches or by prioritize AMS tickets in this aspect. Sometimes this even meant to disable validations which would actually be correct, but are not followed by ZAS itself and can block a business process.

In order to supervise all data import and export batches I provided an extensive monitoring SQL to the IK team and also to IGS batch operation. This enabled both parties to have a common understanding of the system status and a discussion base for the way forward in case of aborted batch runs.

But also SVA-internally the data flow had to be monitored. Actually, IK can also be described as a cross application, as it is the main input provider for the pension (RE) business and some legacy applications. Knowing about these dependencies is essential for having a stable IK business - and at the end correctly paid-out insureds.

(1933 of 5000 characters)

Solution Input to Winning Bids (CC8)

Created the specialty or technical focus area related solution within winning bids, proposals, or contract extensions.

Provide two instances that demonstrate you have met this criterion.

Guidance for this capability statement

Your examples should show you created the specialty or technical focus of a solution, alone or with mentoring or supervision.

Example 1: VISTA BW Proposal

Duration:* From 11 Jul 2011 To 31 Aug 2011

Project or major activity:* VISTA BW Proposal

Description:*

As described earlier, VISTA was initially planned to be deployed in three phases (E1 - E3). After several delivery issues and delays it was decided to discontinue the phase 2 and 3 applications, freeze the development and ramp down these modules in 2008. The module "Contribution" (Beitragswesen, BW) was one of the affected application.

In July 2011, IGS has asked IBM to provide an offer and solution approach for finishing BW. The target of the solution should be to replace the current BW application on the legacy system and to have it integrated into the VISTA context. The offer should cover the following points:

- Effort estimation for the BW legacy replacement of 4 sub modules
- Cost estimation for implementation and support till go-live (SIT, PPT, rollout etc.)
- Schedule for scope refinement, ramp-up, delivery and testing
- Assumptions and preconditions
- Technical solution outline and variants

I joined this proposal to draft a technical solution for integrating a submodule of BW into VISTA. It matched my experiences in data replication functionality (like the Back Interface BI in VISTA), as a similar approach was to be considered for this proposal.

To get a basic understanding of BW and to identify the required components for the new solution, I got in contact with a former BW architect. With his input, I worked out a possible solution (plus an alternative), let it review by the leading central architect, estimated the required effort and integrated my part into the final presentation and estimation sheet.

(1518 of 5000 characters)

Example 2: VISTA PV + IK: Suggestions for Changes

Duration:* From 01 May 2012 To 30 Sep 2013

Project or major activity:* VISTA PV + IK: Suggestions for Changes

Description:*

In May 2012 the IBM AMS VISTA Management asked all Architects to provide suggestions for new changes which should extend or improve the current system - and to generate new business for IBM.

I was able to provide the following list of possible changes:

PV

- Extended support for foreign banks and bank fusions
- Usability improvements for French and Italian speaking users
- Redundancy cleanup redesign / finalization
- Extended validity checks for historized objects

IK

- Printing fallback mechanism in case of missing employer information
- Provide letter salutation in case of documents for non-natural persons
- Improve the handling of multiple-assigned billing numbers
- Redesign crucial batches to improve stability and performance

The total estimated effort (i.e. business for IBM) was approximately 100 person days.

In the meantime, some of these changes were accepted and assigned to IBM. I continue to provide further suggestions for changes for each AMS ticket which describes a problem but which is not a defect per definition.

(1010 of 5000 characters)

IBM Business (CC11)

Demonstrated a keen understanding of the offerings of your business unit as well as a cursory understanding of the services and products available in other areas of the company.

Provide two examples that show how you used your knowledge of your business unit's offerings and services or products in other areas of IBM to meet a client need.

Guidance for this capability statement

Think about when you worked on projects (sales, services or support) that crossed business units, did you explain what your organization did to others on the project? Did you learn more about IBM's structure, offerings or products? You should be able to explain the offerings of your business unit to clients, business partners and team members. In addition, you should be able to explain at a general level the services and products available in other areas of the company. You should know how to find resources within IBM or IBM Business Partners to solve client business problems that cross IBM department and divisional solution areas.

Example 1: General IBM GBS Know-How

Duration:* From 01 Apr 2007 To 30 Sep 2013

Project or major activity:* General IBM GBS Know-How

Description:

As a member of GBS Switzerland I attended regular meetings (All Hands Meetings, AMS Roundtables) to be informed about the organization structure and business updates from the Service line.

I learned that GBS is one of the two major divisions of IBM Global Services - GTS is the other one. IBM GBS is the world's largest IT consultancy service firm and accounts for around 1/5th of IBM's global revenue. It is currently divided into seven service lines: Strategy & Transformation (S&T), Business Analytics and Optimization (BAO), Application Innovation Services (AIS), Application Management Services (AMS), Mid Market, SAP Applications and Microsoft Applications.

I followed organizational changes with great interest, e.g. when IBM Switzerland and IBM Austria together formed the "IMT ALPS", which was later on consolidated with IBM Germany into a new "IMT DACH".

Within VISTA I could collaborate with colleagues from other service lines and countries (Germany, Austria, India), and could also work on a typical project handover from AIS (development) to AMS (maintenance).

(1071 of 5000 characters)

Example 2: IBM Global Delivery

Duration:* From 01 May 2007 To 30 Sep 2013

Project or major activity:* IBM Global Delivery

Description:

IBM is a leading company in Global Delivery (GD). As a worldwide operating enterprise, IBM makes heavy use of outsourcing parts of projects and business into countries with lower cost rates. An additional benefit is the huge pool of employees with skills in almost every relevant topic. IBM has delivery centers in Brasil, China, India, Romania and further locations.

Organizing a GD project is challenging as many GD troubled project showed in the past. Very often the root cause are cultural differences and resulting gaps in expectations, communication, quality and delivery excellence.

In order to address these problems, IBM Switzerland initiated a new approach with the "India Program" in 2007:

- Send a number of Swiss application developers to India for around one year
- Work in Indian teams as "normal" application developers
- Learn about India
- Return to Switzerland
- Take over leading roles in projects with India
- Positive effect for customers and IBM

I was one of the 11 assignees to participate in the India Program. After some cultural awareness courses we got technical training in Kolkata, India, together with Indian new hires. Then I was assigned to a GD project by MAERSK, Denmark in a Java developer role. In this time I could gain hands-on experience about GD methods and approaches, challenges, solutions and last but not least about the Indian culture.

After returning to Switzerland, I was co-author in the Swiss GD community (https://w3.tap.ibm.com/w3ki07/login.action?os_destination=%2Fdisplay%2FGDCS - not online anymore) where we documented our experiences and best-practices for future GD projects. These experiences also helped me to work in the GD setup of the VISTA project.

(1690 of 5000 characters)

Solution Development (CC12)

Created the structures of a solution that was validated to meet one or more business or technical requirements.

Document two instances that demonstrate you have met this criterion. The solution must have been communicated to the development team and reviewed or validated by the client (internal or external). Provide only instances where your role was as the lead for your technical specialty within the project or for a significant subsystem or component.

Guidance for this capability statement

Creating the structures of the solution includes documenting it, for example, with UML or another modeling notation that can then be validated with the client and project teams. In a sales situation, this includes proposing products or solutions that meet or exceed the client business requirements.

Example 1: VISTA PV - IBAN in foreign countries

Duration:* From 01 Mar 2012 To 01 Jul 2012

Project or major activity:* VISTA PV - IBAN in foreign countries

Description:*

In the rollout phase of VISTA, the mandator Liechtenstein (LI) came quite late with the requirement to support payments to foreign accounts. This change required some fundamental modifications in the program logic. As lead architect for PV it was my job to deliver a working solution on time, i.e. before go-live of LI.

First, I set the limits of the scope for this change: LI, IGS and me agreed to support foreign payments of type IBAN only. Then I met with the PV product manager to draft the solution and finalize the specification. On Wikipedia I studied the different formatting definitions of an IBAN in different countries. This was one of the key problems: the current implementation used an automatic lookup to get a handle to an existing bank object in PV via the encoded clearing number in each Swiss IBAN (the digits 5 to 9 of a Swiss IBAN define the clearing number of the hosting bank). This is different for other countries, so it was clear to me to parametrize the solution.

I put the IBAN definitions into the VISTA code management structure which would eliminate hard-coded entries. The RDC developers could then implement the changed logic of the bank lookup mechanism based on the IBAN. In a first build we could test the solution, and it was working fine within PV. But the first system integration tests (SITs) with the billing system (Abrechnungssystem, AS) and extended analysis showed some incompatibilities with SAP. So I sat together with the AS team; we found out that some information on the Wikipedia page was wrong and that the PV method for the lookup of banks has to be extended.

The PV product manager from IGS and I discussed a possible solution. My suggestion to re-use a manual lookup of banks for some countries was accepted and communicated to LI. I updated the specification and configuration and instructed the PV developers and testers accordingly.

As we were working on a new requirement IGS assigned a new change to IBM, which could be delivered on an interim release (iChange) - which was late, but still on time and without following Defects.

(2084 of 5000 characters)

Example 2: VISTA PV - Notification

Duration:* From 01 Jul 2008 To 01 Jul 2011

Project or major activity:* VISTA PV - Notification

Description:*

One big benefit of VISTA compared to the legacy system was the usage of business roles and the extensibility on them. A first concept to inform the business applications if something relevant changed on an insurant's basic data was partially implemented in an early project

phase already. However, it soon became obvious that this first approach would be too complex to understand, use, implement and maintain. It was therefore decided to start from scratch.

I started by interviewing the subject matter experts (SMEs) and lead architects of the business applications about their requirement. This was helpful but also difficult, as these applications were under development, resulting in many open points. However I could already outline the new concept in UML diagrams, use cases, interface definitions and in a technical overview document. In this phase I already involved the lead developer from RDC who also shared his ideas. The central architecture board finally reviewed it closely. After some rework I wrote an additional document which reflected the business side. I could present a short overview to the customer, who gave the green light for implementation.

I assigned individual features for the DB team, batch team and the PV developers and testers. Defining the code management configuration was my job and also a precondition for further steps. After some development builds all features were implemented and we could start testing - at least to the point where the notifications were created. The business applications started later with their implementation of the notification handling, which meant some supporting activities of this concept.

In 2011 the first business applications could completely get rid of the former manual "notifications" of PV and many time-taking tasks were completely automated.

(1820 of 5000 characters)

Problem Resolution and Analysis (CC13)

Performed logical analysis and problem solving.

Provide two instances where you demonstrated the ability to perform logical analysis and problem solving. Provide a short description of the circumstances and the skills that you used to succeed.

Guidance for this capability statement

Your examples should show how you solve problems and provide analysis that supports your assigned tasks, and do so without guidance in routine situations. For more complex situations, mentoring or supervision may be used.

Example 1: VISTA PV - Ticket Analysis for BI

Duration:* From 01 Jul 2008 To 31 Dec 2010

Project or major activity:* VISTA PV - Ticket Analysis for BI

Description:*

Since go-live of phase E1a I was involved in VISTA AMS, which support and maintain all VISTA mandators. During E1a, the SVA's dependency on legacy data was much higher than in phase E1b, as all business applications used the (replicated) basic data from PV legacy. Therefore we put high attention on the VISTA Back Interface (BI) which replicates VISTA PV data to the legacy system (EDV2000).

BI works as a combination of DB2 processes, interface tables and regular batches on VISTA and legacy side. Some small differences in the business rules between VISTA and EDV2000, certain data constellations and some open defects prevented a proper replication in certain situations. In this cases, event tickets were created automatically, indicating some lines of log

files only, though.

I was responsible to analyze these events in detail and initiate the required follow-up steps: data corrections as RfS (Request for Service), create defects or request changes or providing possible workarounds. For the analysis part I had read-access to the productive VISTA databases. This allowed to analyze the current data constellation as well as its history (journal tables). I could however not access the EDV2000 database, so I could not see what finally arrived on the legacy system via SQL. As a workaround we had restricted VISTA user accounts, allowing us to create BI reports with the client. These reports gathered data from VISTA and EDV2000, showing the connection and to some extent also the errors.

After some time I found many similar tickets and resolutions for them. In most cases, an SQL had to be included in a template, which was then executed by the DB2 provider (T-Systems). But even though I gained experience and speed in ticket resolving, the increasing amount of mandators and some cases which requested corrections on VISTA and EDV2000 side resulted in an increasing backlog of automatic and manual tickets.

To improve the ticket situation I organized workshops with the IGS DB administrators of EDV2000. This reduced some dependencies and delays, as the result of a data correction could immediately be verified. The productivity in these workshops was great and resulted in a high client satisfaction.

(2212 of 5000 characters)

Example 2: VISTA IK - Problem Analysis

Duration:* From 01 Sep 2011 To 30 Sep 2013

Project or major activity:* VISTA IK - Problem Analysis

Description:

With go-live of phase E1b also the application IK had to be supported by IBM AMS. Similar to the experiences we made in E1a (BI / PV), we generally continued with the same processes, but more ITIL based. Only a limited time should be spent for a first "diagnosis" of an incident tickets for an initial categorization (known or unknown problem). Based on this first action we requested (for unknown problems) problem tickets for a deeper root cause analysis.

One typical example for the required tasks was a group of tickets, opened by the customer based on the observation that generated documents were not archived in the external system (ELAR Advanced). The general flow of documents is:

- The business application (e.g. IK) creates a request for the Output System (OS)
- OS calls the data extractor class for the specified document to retrieve the business data
- The business data is combined with the template elements to a complete document (PDF)
- The ELAR Interface (EI) is called to initiate the transfer and the archiving of the final document in ELAR

Now the ELAR administrator observed that many documents ran on errors and were not successfully finalized. For each document type he requested an analysis and correction measurements. An initial analysis started in EI, showing that some mandatory fields haven't been set. The root cause for this had to be found in IK, though.

I verified this initial position with queries over the EI tables, to get the keys to the OS request data. OS is structured in a rather complex way (key-value pairs for each element of a document), so I first needed to filter and structure the data in a readable form. The SQL queries got more complex, but also more configurable, resulting in quick summaries of the

data constellation in VISTA, OS and EI.

From these results I could conclude that the problem was invalid partner data of referenced employers. I could conclude the analysis by requesting RfS for the data corrections and by suggesting a change which would allow an adapted logic to handle with such data constellations.

(2057 of 5000 characters)

Personal Impact (CC14)

Demonstrated understanding of the relationship of the personal contribution to the context of the overall objective of the project or engagement.

Describe two instances of your personal contribution in relation to the context of the overall objective of the project or engagement.

Guidance for this capability statement

Provide a short description of the nature of your contribution, the technical skills that you used to make your contribution and the relation between your contribution to the overall objective of the project or engagement.

Example 1: VISTA PV - Duplicate NNSS

Duration:* From 01 Dec 2012 To 01 Feb 2013

Project or major activity:* VISTA PV - Duplicate NNSS

Description:

A long-lasting problem in VISTA PV were duplicate partners. To some extent, duplicates have been created by user errors or migration issues, but the main root cause were automatic reconciliations from other applications. With the introduction of the new Swiss social security number (NNSS) the reconciliation batches of the application IK and IPV relied on the uniqueness of an NNSS over all partners. But some special concepts of IPV and IK and a specification fault enabled these batches to generate hundreds to thousands of duplicate partners inside PV.

For the SVAs it was very difficult to handle duplicate partners: sometimes, the business roles for one partner were distributed to two different partners. In other cases, one insurance business (respectively business roles) was assigned twice to one NNSS (but two identical partners). This led to duplicate administrative work, in the worst case these insureds received letters or even payments twice. Often, a manual cleanup was difficult or even impossible.

After the root causes had been fixed, I was asked to provide a solution concept for a data cleanup. This concept consisted of two mechanical iterations, considering the distributions of the business roles. The remaining amount should then be communicated in a list form to the SVAs manual data cleanup. The customer approved the concept and also asked for a time schedule.

The script had to distinguish between multiple scenarios, each iteration consisted of many sub tasks and control steps. Also, the effect on follow-up batches had to be considered. For the biggest mandator (SVA Zurich), after the two initial iterations I discovered an additional possibility for a reduction of duplicates, which reduced the amount again significantly.

The scripts ran successfully on all mandator's systems. In Zurich, the duplicates could be reduced from around 25.000 to approximately 300, which was - after I provided a detailed list - easy to handle for SVA employees.

(1977 of 5000 characters)

Example 2: VISTA PV - Special Effort for urgent Queries

Duration:* From 18 May 2011 To 25 May 2011

Project or major activity:* VISTA PV - Special Effort for urgent Queries

Description:*

VISTA was in the phase of "accelerated rollouts" when some urgent pre-migration tasks of one mandator (SVA TG) were still open. TG was already live with VISTA phase E1a (basically PV only) in which the data dependencies for the business applications (on legacy) were not that strong. For a successful go-live of SVA TG the PV data on production had to be completed first with three tasks:

- Assign correct civil-state-since dates to all partners
- Assign correct country of living-since dates to all partners
- Assign correct NNSS (new social security number) to all partners

As input, the customer could only provide a list of IDs and values from legacy only. I had to map these values to vista data, provide SQL scripts for importing the modified list files and the follow-up processing. Like for other mass updates in PV I had to consider the effects on follow-up replication batches which could not handle updates in too big portions and find a way to split the workload. The real challenge however was the time. The migration team could not provide the input files before the "work freeze" before go-live, and the script had to run before some other migration tasks.

A special effort from my side over nights and the weekend was required to deliver the solution. Fortunately, all three scripts have been executed successfully and on time. The project manager from the customer side sent an appreciation mail to me afterwards (see file [compliment_steger_spezialeffort_20110524.png](#)).

(1475 of 5000 characters)

[compliment_steger_spezialeffort_20110524](#) (PNG, 33 KB, 06 Oct 2013)

Interface to Architecture (CC15)

Demonstrated understanding of the relationship of the personal contribution to the context of the enterprise or project architecture. Engaged mentor, as needed, to clarify the relationship between personal contribution and the architecture.

Describe two instances that demonstrate you have met this criterion. Provide a short description of the nature of your contribution, the technical skills that you used to make your contribution and the relationship of your contribution to the enterprise or project architecture.

Guidance for this capability statement

You need to understand the relationship of your work to the overall client environment. You must be able to describe how the architecture functions, and where your contribution fits into the whole. Remember that there are many types of architecture - hardware (e.g. RISC), operating system (e.g. Linux), middleware (e.g. WebSphere Portal), frameworks (e.g. Eclipse), model (e.g. Service Oriented Architecture).

Example 1: VISTA PV - Data Synchronization to Legacy and External Systems

Duration:* From 01 Apr 2008 To 30 Sep 2013

Project or major activity:*

VISTA PV - Data Synchronization to Legacy and External Systems

Description:*

PV is a cross application in the VISTA context, having 276 Interfaces for general usage or customized for the business applications. PV is an important data provider for them and the dependencies have to be considered and understood both in development and operation.

During the development phases of VISTA it was important to maintain the the interface overview documents. The interfaces have been structured into several groups, mainly according their assigned business applications. This allowed to use very specific functionality which could be defined and maintained by their owners without interfering others. Obviously, also many interfaces were specified for general usage which were used by all applications.

Working for some Back Interface (BI) Changes and while analyzing AMS incidents helped me to better understand the system relationship. PV data is not only used by other VISTA applications, but also of great importance for the legacy system. For example, if a payment address could not be replicated to EDV2000 the insurant could not be paid out.

Different replication mechanisms had to be applied for similar reasons to other systems like SAP, ELAR Advanced and later on GILAI. Being involved in the development of these system bridges was a big benefit for later addendums and support activities.

However, the most visible example for PVs system context is probably the notification concept which I made the specification for. During interviews with SMEs and later on in the testing phases I could closely follow the effect of a relationship update on a partner in PV and the consequences in the pension application.

PV is a core component in the VISTA context, which itself is only one component of a very complex system context. This fact however does not diminish its importance.

(1796 of 5000 characters)

Example 2: VISTA IK - Data Exchange with other Customers

Duration:* From 01 Sep 2011 To 30 Sep 2013

Project or major activity:*

VISTA IK - Data Exchange with other Customers

Description:*

As already described in (CC7), IK can also be described as a cross application as a main input data provider for the pension (RE) business, some legacy applications and other SVAs. In IK it was essential to keep the system environment in mind.

The data exchange via the central compensation fund (Zentrale Ausgleichsstelle, ZAS) is

probably the most apparent part in IK to demonstrate the interface tasks of the application to the entire business context. Simplified, IK communicates with the ZAS Assistant (ZA), which is responsible to format, validate and transform VISTA data into the XML format in interface tables of the ZAS Collector (ZAS Kollektor, ZK). From the ZK tables, the XML structures are imported and exported from/to ZAS, which is outside of the VISTA context, but the central element in the business flow for the information flow between the SVAs.

I have worked with many different XML structures and verified in some cases the compliance of VISTA with ZAS. I defined business rules to cut some VISTA data structures to a certain amount of characters in order to be able to export booking information without validation errors. It was instructive to see that a too long name for an employer could be the reason for validation errors, unprocessed transmission to ZAS and therefore pensions which could be calculated for an insurant, whose business is actually at another non-VISTA SVA.

Similar attention had to be assured when I extended the data definitions for data exchange from/to Legacy, e.g. for the Contributions application (Beitragswesen, BW) in EDV2000.

Specifying Changes for IK always meant to check the implications on other VISTA applications, legacy and ZAS.

(1685 of 5000 characters)

Asset Re-use and Harvesting (CC16)

Reused, when appropriate and cost effective, assets, objects and materials.

Provide two instances that demonstrate you have met this criterion. Provide a short description of the nature of the problem, the build or re-use options you considered, the rationale for your choice and the consequences of your decision.

Guidance for this capability statement

Your examples should show how you found and reused existing objects or materials to complete assigned project tasks on client projects and engagements. When more than one asset applies, you determined which asset was the most useful in the situation. You defended that decision to others and explained the benefits (or drawbacks) of the asset.

Example 1: VISTA PV - Use of DB2 Functions

Duration:* From 04 Apr 2013 To 05 Apr 2013

Project or major activity:* VISTA PV - Use of DB2 Functions

Description:

Problem description:

In VISTA, partners are either created manually or via some reconciliation batches. Around 20% are imported from an external source (ZAS). In this case the name of an insurant is stored in capital letters in PV. With the inclusion of external applications the customer requested to change these cases in proper capitalization. However, the DB2-internal function INITCAP could not be used, as this function was introduced with DB2 9.7 only, but VISTA was using version 9.5 at that time.

Options:

A research showed the possibility to use User Defined Functions (UDF) in DB2. A search for existing assets resulted in two ready-to-use files:

IBM: <http://www.ibm.com/developerworks/data/library/samples/db2/0205udfs/>

External: <http://www.zinox.com/node/13>

Decision:

Both assets would serve the same purpose, but the version from IBM provided more samples and seemed to be used more frequently. Therefore I decided to go for this asset which was working perfectly.

(971 of 5000 characters)

Example 2: VISTA IK - Quick Ticket Analysis Sheet

Duration:* From 01 Nov 2012 To 01 Feb 2013

Project or major activity:* VISTA IK - Quick Ticket Analysis Sheet

Description:*

Problem Description:

The application leader of IK created an Excel sheet to track AMS tickets and combine them with known changes and defects. In this absence and after he had left the project I took over this asset for further ticket tracking.

Assets:

Kategorisierung neue Tickets 2012-10-02_GB.xls

Purpose:

With this work product I could compare new tickets against existing ones. I used it for stand-up meetings with the team to discuss, categorize and assign new tickets. Ticket data could be copy-pasted from the ticket dump, one worksheet checked for duplicate entries. Also, it allowed a quick way to search for known problems respectively existing defects or changes.

(670 of 5000 characters)

Verbal Communication (CC2)

Created and delivered compelling verbal communications in an organized and persuasive manner, adapted style for different audiences in project deliverables and in a client environment.

*** List two presentations that were published or provided to internal or external clients and that demonstrate your ability to effectively communicate decisions and designs in your technical specialty. For each presentation, provide the name of each presentation, a short description of the purpose and the outcome of the discussion or presentation.

Guidance for this capability statement

Your examples should show how you communicate your technical expertise orally. Try to be specific in both the actions and results. ***THE FOLLOWING IS AN EXAMPLE*** "I delivered a 2 hour presentation to XXX to explain YYY. I held a Questions & Answers session to address....:"

Example 1: IT Architect in a GD environment - presentation for IBM apprentices

Duration:* From 05 Sep 2012 To 05 Sep 2012

Project or major activity:*

IT Architect in a GD environment - presentation for IBM apprentices

Description:*

Filename:
VISTA_Architectural-Workflow_Apprentice_20120905.ppt

Purpose:
To demonstrate the job role "IT Architect in a GD environment" as part of a day for IBM's apprentices I held a 2 hour presentation to 3 IT apprentices. The target was to give them an impression how a typical day or task could look in a Global Delivery (GD) project.

(334 of 5000 characters)

[VISTA_Architectural-Workflow_Apprentice_20120905_compressed](#) (PPT, 2.5 MB, 13 Aug 2013)

Example 2: VISTA PV - Education Sessions for Product Managers

Duration:* From 14 Jun 2011 To 14 Jun 2011

Project or major activity:* VISTA PV - Education Sessions for Product Managers

Description:*

Filename:
IGS Training VISTA PV Notification_20110616_compressed.ppt

Purpose:
For internal training for the product managers, the customer requested that SMEs or lead architects would hold an education session. In this presentation I explained the new topic of the PV notification concept. The presentation file should be archived and therefore of a high quality regarding the comprehensibility. All the information on these slides should be correct, but simplifications should be used for complex topics. Therefore I used many diagrams instead of specification details and added some more technical details in a backup section only.

(630 of 5000 characters)

[IGS Training VISTA PV Notification_20110616_compressed](#) (PPT, 1.5 MB, 13 Aug 2013)

Example 3: VISTA PV - Migration Problem Description

Duration: From 28 Jan 2009 To 28 Jan 2009

Project or major activity: VISTA PV - Migration Problem Description

Description:

Filename:
VISTA_PV_Zahlungsanweisungen_Adressen_0.4_d.ppt

Purpose:
After many mails from and to the migration team (IGS and IBM) addressing a problem regarding a migration issue with payment addresses, I decided to hold a presentation for all involved parties, describe the problem, propose a solution and target to define a way forward. This session aligned all parties and for future incidents on this topic the agreed process

could be applied.

(444 of 5000 characters)

[VISTA_PV_Zahlungsanweisungen_Adressen_0.4_d](#) (PPT, 346 KB, 13 Aug 2013)

Technical Support capability statements	Examples	Status required	Action
Solution for Client (CFSup7)	2 required	2 completed	View evidence...
New Business Opportunities (CFSup6)	2 required	2 completed	View evidence...
Change Management (CFSup5)	2 required	2 completed	View evidence...
Client Deliverables (CFSup4)	2 required	2 completed	View evidence...
Technical Leadership (CFSup3)	2 required	2 completed	View evidence...
Problem Management (CFSup2)	2 required	2 completed	View evidence...
Client Technical Advisor (CFSup1)	2 required	2 completed	View evidence...

Solution for Client (CFSup7)

Proposed a successful solution or part solution to a non-trivial client problem within your technical specialty.

Provide two instances that demonstrate you have met this criterion. In each case, please describe briefly the client's problem and the solution you proposed.

Guidance for this capability statement

This is about creating a solution proposal. Just because you're in Support instead of Sales doesn't mean you never have to create a proposal. Let's say your specialty is Storage. You could write something like this: "On the Acme project, the client had had several old file servers that were increasingly starting to fail, resulting in frequent unplanned outages and long backup times. After conducting some research, I created a proposal to replace them with a DiskMasterTronic 5000 Turbo using RAID99 technology. I presented the proposal to the senior management team, which was hesitant at first. I performed a cost benefit analysis, comparing the investment cost to the maintenance cost of the current system, and demonstrated to the client they would recoup the cost within 3 years. The proposal was accepted and the project was implemented on schedule." Focus on the proposal part here, don't go too deep into the actual technical solution. Also, your contribution doesn't need to solve the entire problem. Let's say the client wanted to increase maintenance costs across the board (storage, network, help desk calls, servers, software licenses). Cutting storage costs would be a "partial solution" that fits within your specialty (in this case Storage). The problem can't be trivial. Writing a shell script to automatically check if a file system is > 90% full is not sufficient.

Example 1: VISTA IK - Merge IK Mini Partners

Duration:* From 01 Jun 2010 To 30 Jun 2010

Project or major activity:* VISTA IK - Merge IK Mini Partners

Description:*

The Scope Refinement phase in VISTA was a very collaborative and innovative time. Suggestions for changes could be submitted easily on a SharePoint team room where they were reviewed by a "Change Management Board" (CMB). In parallel, phase E1a was already live and some experiences from production could already be included in such proposals.

I saw that duplicate partners would lead to massive problems and high efforts to cleanup wrongly assigned business roles. Besides these high efforts also the financial impacts in case of multiple paid-out compensations could have been critical. Therefore I decided to propose a solution in which the SVAs could resolve such issues by their own without any requests for service.

IK knows the concept of "mini partners" which are actually normal PV partners created by IK, having no address and an IK business role only. They are created if no existing partner can be found based on the social security number (SSN). However, many partners didn't have an SSN or only an old one. This led to the fact that most duplicates consist of a "normal" partner with any business roles and an IK mini partner.

My proposal described a concept, how such partners could safely be merged: the IK role and the SSN of the mini partner would be attached to another partner (often having already different business roles), the SSN deleted from the mini partner and finally the mini partner would be terminated.

For capacity and prioritization reasons the proposal was not approved by the CMB, but the concept was not done in vain: parts of it could be re-used later for data cleanup activities.

(1616 of 5000 characters)

Example 2: VISTA PV - Force Replication button to external System

Duration:* From 01 Dec 2012 To 31 Dec 2012

Project or major activity:*

VISTA PV - Force Replication button to external System

Description:*

My concept of the integration of the external system GILAI into VISTA (see CC3) was accepted by the customer and in development. This batch-driven solution could however not guarantee an "immediate" replication of VISTA data to GILAI. For technical reasons we could only guarantee to run the replication batch every 5 minutes, but even this time could not be guaranteed in case of high workloads. The 5 minutes maximal replication time was accepted but not optimal and compromise only.

Then I thought about the possibility to trigger the replication manually. It was however not possible to start a batch on demand as this would interfere with the existing Batch Job Plan (BJC). But when I had another view at the process chart I found that GILAI only consumes data from some target tables and is not dependent on any processed steps before.

I made a short proposal about including a button on the GUI which would then trigger a "shortened" replication to the target tables, circumventing all batch dependencies. The attributes of the currently active partner would be written to the target tables, ready to get captured by processes of GILAI.

The customer liked this approach for two reasons: cost efficiency by using already existing tables and services from the initial concept and the removal of batch dependencies and to offer a guaranteed "immediate" replication if required. IGS handed over a new change feature to IBM for implementation which has successfully been delivered.

(1480 of 5000 characters)

New Business Opportunities (CFSup6)

Identified sales opportunities by recognizing and articulating potential new business opportunities related to clients or client relationships.

Provide two instances that demonstrate you have met this criterion. In each case, please describe briefly the circumstances and the potential opportunity and how you articulated it.

Guidance for this capability statement

A very common scenario is where you as the support provider make a recommendation on how to improve an existing system that is expensive to operate, or prone to errors, or difficult to support. IBM gets most of its business from established accounts, and very few people understand the client's needs better than those supporting the solution.

Example 1: VISTA PV - Integration of Audit and Analysis Functionality

Duration:* From 01 Mar 2010 To 01 May 2011

Project or major activity:*

VISTA PV - Integration of Audit and Analysis Functionality

Description:*

In PV there are important partners like SVA branch offices whose addresses and payment addresses are referenced by many partners for payments or

correspondence. Changes on such objects are therefore crucial and should be checked before a payment process. Accordingly, before each bigger payment run I observed increasing AMS tickets requesting information like "How many insureds have a payment relation with branch office X?" or "Please send a list with all addresses which are references by more than 20 insureds and which were changed during the last 2 months."

In some cases we could not provide the list on time. Also, each SVA wanted such lists minimally different and there was no standard process for this tasks. I thought about integrating such a reporting functionality into VISTA, thus eliminating the dependencies of the SVAs on other parties and offering the possibility to create such reports on demand.

One VISTA module offered the possibility to create printable documents: List Generator (LG). LG was mainly foreseen for the business applications and not used by PV at all. However, LG is application independent and can be called from every application. I suggested to define a template for the most important attributes and integrate a new list type for PV into LG.

As LG offers a parametrization for each list, I could make a small proposal for a change which would meet the different requirements from each mandator. The costs for this change would be equalized after a few months already. IGS assigned my proposal as new change to IBM, offering a good service for its mandators and reducing IBMs effort for AMS tickets.

(1641 of 5000 characters)

Example 2: VISTA PV - Replication of Banks to Legacy

Duration:* From 01 Sep 2009 To 01 Nov 2009

Project or major activity:* VISTA PV - Replication of Banks to Legacy

Description:*

Banks and payment addresses are very important objects in VISTA, as paying out the insured persons is one of the core business of the SVAs. As the SVAs still use some legacy systems for some business applications, all payment relevant data has to be replicated from VISTA to legacy.

Replicating bank basic data was not in the initial scope of the BackInterface (BI). A complex manual process had to be established in case new banks had to be defined in the system: the SVA had to create the bank in VISTA, create an AMS ticket with the details, IBM AMS then prepared and forwarded a template to two legacy teams which also had to create the new bank in legacy applications. They returned some database keys from legacy, IBM updated the form, sent it to the Service Desk for handing over the form to the hosting provider for executing an SQL which created link (keymapping) between VISTA and EDV2000.

Even though the steps were not complex, the total execution time for this process was easily two working days. Usually requests for new banks came just before payment runs, therefore these tasks were often time-critical.

I suggested to analyze the technical requirements and provide an effort estimation for integrating banks into the BI replication. The customer was happy to have the possibility to compare the costs and assigned an analysis change to me. For this I drafted the technical solution and estimated the efforts in detail. However, the efforts were too high for the customer (he assumed that BI will be a temporary solution only for a relive short time), so the feature had never been implemented.

Nevertheless, the Analysis was paid and I could later optimize the manual bank process at least - and even outsource the IBM part of it to RDC India.

(1758 of 5000 characters)

Change Management (CFSup5)

Managed change with significant technical scope or business impact.

Provide two instances that demonstrate you have met this criterion. In each case, please describe briefly the circumstances and the relevant aspects of change management involved, such as impact analysis, prioritization, tracking and history, use of tools, emergency response, testing, roll-out, or contingency planning.

Guidance for this capability statement

Examples include: ·Ensure all changes are authorized and reviewed for their potential impact· ·Give personal attention to high priority/emergency change processing· ·Ensure all changes are tracked and that history is available.· ·Track and authorize changes using appropriate tools· ·Allow emergency changes to be made by authorized personnel· ·Have a back out and recovery plan in place for major changes· ·Define go, no-go decision points..

Example 1: VISTA PV - Integration of IBAN

Duration:* From 01 Sep 2008 To 01 Nov 2008

Project or major activity:* VISTA PV - Integration of IBAN

Description:*

In the conceptional phase of VISTA the International Bank Account Number (IBAN) was not yet in use in Switzerland. Whereas early Changes for PV introduced this payment type in VISTA, it had to be disabled as this type was unknown in legacy (EDV2000) and such payment addresses could not be replicated via the Back Interface (BI) to EDV2000.

When the Swiss Post announced to use IBAN as default payment type and charge CHF 5.- for each transaction to non-IBAN accounts, the need for an integration of this new payment type in PV, BI and EDV2000 became a high priority change request; the financial impact on the SVAs when not implementing this change was obvious.

I split the change into two separate features, one for PV and one for BI. But first the solution had to be outlined on a high detail level as I had to consider the needed modifications in EDV2000, too. BI uses fixed-length strings in interface tables which are polled by legacy batches. I defined a modified string format and let it review by the Architect for EDV2000.

At the same time I specified new interface methods for PV and BI to enable the transfer process for IBAN payment addresses. Also, VISTA code Management (CM) values and hidden GUIs had to be enabled again. More complex however was to define new validation rules and a logic for retrieving the correct bank basic data to an IBAN.

The concept for these parts were reviewed by the leading central architect of VISTA and all involved developers. Then, the implementation could start: DB and CM Changes were targeted for the same sub release, completing all preconditions for the model update, GUI modifications and the new Java logic.

For this change, an early testing involvement of the legacy side was important and helpful: defects could be found and fixed in an early pre-build.

(1805 of 5000 characters)

Example 2: VISTA IK - Include Correspondence Addressee in legacy imports**Duration:*** From 01 Dec 2011 To 01 Feb 2012**Project or major activity:***

VISTA IK - Include Correspondence Addressee in legacy imports

Description:*

Because the application "Contributions" (Beitragswesen, BW) is not integrated in VISTA but is still operated on Legacy, the BW employer roles incl. correspondence addresses are missing in VISTA PV. In Legacy, these information are stored in a table. While VISTA batches import employer data, the connection to the correspondence address in PV is missing. The customer asked to have this information imported, too.

For this change I made an impact analysis first: I listed all business rules, GUIs, Interfaces and processes which would be involved. Based on this I could make suggestions for the SIT (System Integration Test). A time schedule could also be concluded from this analysis, as several components needed to be in synch in each subrelease, leading to a final build which would fulfill the customer's expectations.

After I assigned all features to the developers and other architects, I tracked the progress on a daily base and updated the current state in a change tracking sheet. I had to handle a fault on the customer's side (the legacy definitions had an error and required updates on the Java code), but this was relatively easy to handle, but required consistent documentation changes in the use case and test data.

We could deliver the change on time, but the very last tests showed a problem when the new parameter was null. This defect required a patch (iFix) as the regular build was already gone. Nevertheless, it was before the Go-Live of this release and had therefore no impact on the production business.

(1526 of 5000 characters)

Client Deliverables (CFSup4)

Created client deliverables.

Provide two instances where you created internal or external client deliverables. In each case, please describe briefly the circumstances, the objective of your work and the nature of the client deliverables you created.

Guidance for this capability statement

Examples include:· Create scenarios and custom demonstrations (e.g. Prototypes, user interface)· Develop custom presentation that represents vendor's products, services and solutions

Example 1: VISTA PV - New form for a standard process**Duration:*** From 01 Aug 2008 To 01 Mar 2012**Project or major activity:***

VISTA PV - New form for a standard process

Description:*

Filename:

50611_ZG_Raiffeisenbank_Dünnerntal_v2.1.doc

Deliverable type:

Standard process form for incident management

Purpose:

As banks were not replicated to EDV2000 (legacy), a process had to be established in case of new banks needed to be introduced or changed in VISTA. This process involved 5 stakeholders and several manual steps. I could establish a standard form which all involved parties could understand and take the appropriate actions. I further optimized it in a later phase and adapted it to a new incident management organization. It was even possible to outsource the IBM tasks to RDC India. The form is still in use for several bank changes per week.

(664 of 5000 characters)

Example 2: VISTA PV - Product Manager Education in Partner Replication

Duration:* From 19 Jun 2011 To 20 Jun 2011

Project or major activity:*

VISTA PV - Product Manager Education in Partner Replication

Description:*

Filename:

IGS Training VISTA PV Replikation.ppt

Deliverable type:

Education material

Purpose:

I was asked to hold a training session for the customer's product managers addressing the replication mechanisms of PV. The presentation should also be usable as a quick reference documentation, internal trainings, on-boarding etc., therefore I used the most important information in the normal presentation slides but added a set of backup slides with further details.

(458 of 5000 characters)

Technical Leadership (CFSup3)

Provided technical leadership.

Provide two instances where you showed technical leadership, for example, leading reviews of impact analysis and fitness for purpose. In each case please describe briefly the nature of the task, the team, the results and the contributions of the team members

Example 1: VISTA PV - Notification for Business Applications

Duration:* From 01 Sep 2011 To 01 Dec 2011

Project or major activity:*

VISTA PV - Notification for Business Applications

Description:*

The time between the implementation of the PV part for the new notification concept and the implementation of the business applications' part was almost 2 years. The business needs for

the customers respectively the business applications were not finalized for a long time. Consequently, before the entire concept could go live, the Architects, SMEs, product managers and end users of the SVAs had to deal with unclear requirements on one hand and a tough schedule on the other hand.

In this phase I got the task to lead a small subproject for the notification concept. My main tasks were to align all teams (PV, RE, HE, EL) to have a common understanding, gather the final requirements from the SVAs and detect open points in the implementation and address them in defects and changes.

I set up a weekly meeting to present the progress to the management and discuss next action points. Usually small sub teams were formed after the meeting to analyze new findings or to test implemented defects or changes. The contribution each team provided were new testcases and updated tracking sheets, from my side I tracked the new defects and changes and a merged tracking sheet for the management.

The new notification concept could go live successfully on time.

(1252 of 5000 characters)

Example 2: VISTA PV & IK - Change Reviews

Duration:* From 01 Mar 2008 To 30 Sep 2013

Project or major activity:* VISTA PV & IK - Change Reviews

Description:*

Reviewing changes was almost a daily work for me in VISTA within PV and IK. The first review usually took place before the real work had started: the customer specified the business requirements in SharePoint for an initial analysis. Besides providing an effort estimation and a technical specification, I had to challenge the customers requirements first. I encountered multiple cases where the business requirements contracted some established functionality or the desired solution would have had negative consequences. In such cases it was good to clarify these points in an early stage already.

For the effort estimation I involved RDC developers who generally used a specific template to calculate the effort. In many cases the outcome however was not realistic, therefore the provided values had to be questioned critically before the final offer could be given to the customer.

Before a change was assigned for development I leaded a handover telco with RDC to clarify open questions and to ensure a common understanding. Of course I was also available for questions which were raised later on. Normally the next review part from my side affected the AIT test cases provided by the RDC test team. This ensured a sensible test coverage before code delivery.

I asked the developers to show the new functionality in a spot check, using phone and remote desktop connections. In this sessions I could get a good impression about the change and already address first findings.

In some cases it was also required to assist the developers with a code review which also explained some troubles the developers or testers were facing.

(1626 of 5000 characters)

Problem Management (CFSup2)

Provided problem management. Managed a significant internal or external client problem from analysis to resolution.

Provide two instances where you managed a significant internal or external client problem from analysis through to resolution. In each case, please describe briefly the nature of the problem, way you approached it, the risk and severity to the client, the resources needed to solve it, involvement of subject matter experts, and how the resolution was implemented.

Guidance for this capability statement

- Apply problem solving skills
- Assess risk and severity
- Locate and allocate resources as necessary
- Involve Subject Matter Experts (SMEs) to solve the problems
- Implement problem resolution (install, test and run patches, upgrades)

Example 1: VISTA PV - AMS Tickets

Duration:* From 01 Feb 2009 To 30 Sep 2013

Project or major activity:* VISTA PV - AMS Tickets

Description:*

Since go-live of VISTA phase E1a I was involved in the AMS incident management process. Generally the customer's tickets came in as incidents. After a short diagnosis (known or unknown problem) we requested problem tickets for deeper analysis. The result then usually led to data cleanup activities (Request for Service, RfS), defects or changes.

For PV I would like to give one example:

Like partners, addresses have to be replicated by the Back Interface (BI) from VISTA PV to Legacy PV. If a partner or address cannot be replicated, for example due to different business rules on VISTA and Legacy, an automatic ticket is triggered during the error logging. In several cases however the user changed an address but the change was not replicated to EDV2000 - but no error had been logged. In the BI reports I saw that all VISTA updates were "processed", but without effect.

At that time I was rather new in the BI context, so I involved the lead architect for BI in the analysis. Together, we found out that the address had been created by the migration and had a special key mapping (a table which combines the technical keys from VISTA and EDV2000). As this address belonged to a payment address only and was only created to establish the same structure both in VISTA and EDV2000, such addresses were defined to be treated in special way by BI: all VISTA updates on the address would be "ignored" and directly marked as processed.

Together with architects from the client we discussed this topic. We had the option either to change the behavior in BI or accept the current state, mark it as "accepted issue" and find a workaround. We decided for the latter, documented the root cause and created a work around instruction document for future cases.

(1752 of 5000 characters)

Example 2: VISTA IK - AMS Tickets

Duration:* From 01 May 2013 To 30 Sep 2013

Project or major activity:* VISTA IK - AMS Tickets

Description:*

For IK, I was involved in the AMS incident management process since go-live of phase E1b. Since beginning, IK had a high rate of incoming incidents, one reason for it was IK's data driven nature and it's status as a cross application. I would like to give one complex problem type for IK as example.

As other applications IK creates documents, which is performed by starting an OS-request (OS = Output system) first, providing some basic data like the name of the template and then let OS call the specific data extractor for the document which fill the template with business data. The Elar Interface (EI) in a later process captures all completed OS requests and send the generated documents (PDF) to ELAR, optionally also into an archive.

An increasing number of OS events, error entries in the EI tables and users complaining that in some cases documents were not printed or archived led to the request for a deeper analysis of the root causes for it. For the IK documents I took over the analysis. I asked the SME for a pre-analysis of the errors for the IK documents, also, I asked the OS architect for some analysis SQL queries. With the input from them I already knew that the problem in 95% was a missing parameter in the OS requests. Without this parameter ELAR could not further process the documents.

This variable is retrieved by PV through a look-up service which returns the correct value if all dependencies can be resolved. Otherwise, null is returned, leading to the missing parameter in the OS request. I crated a complex SQL which linked all dependent objects to one table and could conclude that it was a data problem. In almost all error cases one of the dependencies was broken. For the other cases the dependencies were all correct at the date of analysis. Therefore I extended the query to search for historic data, too, using DB2's journal tables. In these cases, the problematic data constellation was existing at the printing date, resolved later on, but the OS requests were not updated (as such a functionality does not exist).

I could solve the problem tickets by describing the problem, suggested a change for an extended look-up service in IK and providing data correction scripts in follow-up RfS tickets.

(2238 of 5000 characters)

Client Technical Advisor (CFSup1)

Advised and guided the client on technical decisions for the use of vendor products, services and solutions (trusted technical advisor).

Provide two instances where you acted as a trusted technical advisor, advising and guiding the internal or external client on technical decisions for the use of vendor products, services and solutions. In each case, please describe briefly the client needs, a summary of the options you considered, your recommendation and rationale.

Guidance for this capability statement

- Identify problems related to installation, update, configuration, operations or performance·
- Provide subject matter expertise on solution design·
- Provide advice on potential resolutions and their implementation

Example 1: VISTA IK - DB2-Client installation for team

Duration:* From 01 Jan 2013 To 01 Feb 2013

Project or major activity:* VISTA IK - DB2-Client installation for team

Description:*

For the incident analysis in IK, the entire team created and collected a huge amount of SQL queries. It was however not always easy to find, modify and execute the scripts without losing much time.

Together with a colleague from another business application I created analysis scripts which would provide a quick overview of the IK data and some dependent objects by providing a social security number (SSN) only. We made .BAT files which were able to execute an entire chain of SQLs by a single double click and an SSN as input. This enabled the team to have an entire partner report in one file. Additionally, it provided a method to test batches by taking a report before batch execution and afterwards and then comparing the reports side-by-side.

The scripts however required a full DB2 installation which I suggested to all local team members and to the RDC testing team. Before most of the team members used other products like Squirrel or DBVisualizer only. Of course, they are still in use and the DB2 installation is just an addendum - a time-saving one, however.

Some colleagues reported a strange problem when they accessed some tables on a certain release. In a forum I found that the possible root cause was an incompatible DB2 version with some client systems. In this case I asked to install a different DB2 version - which solved the problem.

(1355 of 5000 characters)

Example 2: VISTA PV - Usage of IBM products

Duration:* From 01 Mar 2008 To 07 Aug 2013

Project or major activity:* VISTA PV - Usage of IBM products

Description:*

The VISTA project setup uses many IBM software packages and licenses:

- IBM Rational ClearCase (CC) for project artifacts
- IBM Rational ClearQuest (CQ) for tracking Defects and Changes
- IBM Rational Software Architect (RSA) for modeling and coding
- IBM Rational Manual Tester (RMT) for defining test cases
- IBM Rational ClearQuest Test Manager (CQTM) for tracking the execution of test cases
- IBM DB2 as database application

For my work as architect and application leader I made extensive usage of all these products.

When the test automation subproject started, IBM additionally set up

- IBM Rational Functional Tester (RFT) for designing new automated testcases

During my stay in India for the test automation workshop, I also supported the test architects in capturing the VISTA GUIs and creating correct automation scripts in RFT.

The customer migrates to

- IBM Rational Team Concert (RTC), mainly using it for testing artifacts

I intend to use the web front-end, too for an effective collaboration with the customer.

(998 of 5000 characters)

Application Development and Maintenance capability statements	Examples	Status required	Action
Work products usage (AI11)	2 required	2 completed	View evidence...
Library management tools (AI10)	2 required	2 completed	View evidence...
Design from requirements (AI9)	2 required	2 completed	View evidence...
Methods (AI8)	2 required	2 completed	View evidence...
Configuration management tools (AI7)	2 required	2 completed	View evidence...
Complete solution (AI6)	2 required	2 completed	View evidence...
Development progress and quality tracking (AI5)	2 required	2 completed	View evidence...
Automated development tools (AI4)	2 required	2 completed	View evidence...
Test and debug (AI3)	2 required	2 completed	View evidence...
Programming language selection (AI2)	2 required	2 completed	View evidence...
Application programming (AI1)	2 required	2 completed	View evidence...

Work products usage (AI11)

Demonstrated experience with key work products for Application Development. 1) System Context (APP 011) 2) Logical Data Model (APP 110) 3) Interface Specifications (APP 030) 4) Use Case Model (APP 130) 5) Test Strategy (APP 141) 6) User Interface Conceptual Model (APP 146) 7) Application Maintenance Turnover Package (APP 40153) 8) Interactive Model (APP 160)- User Scenario (APP 305) 9) DFD - Application Model (APP 315) 10) Detailed Gap Analysis (APP 393) 11) Application Programming Interface (ARC 104) 12) Component Model (ARC 108) 13) Non-Functional Requirements (ARC 119) 14) Process Definition (BUS 309).

Provide two instances where you used the key work products to create deliverables. In each case, please briefly describe the circumstances, your role and the deliverables you created. Definition (BUS 309)

Guidance for this capability statement

The work product IDs in parentheses are taken from IBM Unified Method. Equivalent work products defined in other methodology are also valid.

Example 1: VISTA PV - Interface Specifications (APP 030)

Duration:* From 01 Mar 2008 To 30 Sep 2013

Project or major activity:* VISTA PV - Interface Specifications (APP 030)

Description:*

VISTA used the IBM Unified Method and each of its work products. For PV I worked mainly with APP 030 work products.

As previously stated (CC15), PV is a cross application with extensive usage and provision of interfaces. The core documents for the definitions were the interface definitions which were split among the business applications. Each application had its own interfaces, allowing a high level of individuality and independence. I specified and updated many services for the business applications and kept track, which application uses which interfaces.

Of course there were also common interfaces used by many other application. I had to keep them up-to-date and could provide support for the applications about which interface to use and what to consider about them.

In the APP 030 context also the entire PV notification concept found it's place, especially by providing a complex service to analyze the notification entries in detail. As this service provided current and historic PV data and the interpretation of the returned objects was tricky I supported the architects and SMEs in the correct usage when the business applications implemented their part of the notification functionality.

(1203 of 5000 characters)

Example 2: VISTA IK - Use Cases (APP 130) and Business Rules (APP 300)

Duration:* From 01 Oct 2010 To 30 Sep 2013

Project or major activity:*
VISTA IK - Use Cases (APP 130) and Business Rules (APP 300)

Description:*

Like for PV I also worked in IK with many different work product types of the IBM Unified Method. Here I want to focus on the two work products I used most: APP 130 and APP 300.

IK has 221 detail Use Cases in 17 groups. Working on changes, defects and AMS incidents required to consult and update these documents carefully. They were the main tool to decide about defect vs. change, and as architect I had to support the application leader and customer's product manager in such decisions.

Handing over a change to RDC required me first to define the new logic in the specific use cases. Implementation was then based on the use cases and the second main work product group: Business Rules (APP 300).

The business rules clearly decided about go / no go for the user's actions. In IK as a cross application, communicating with external systems and other SVAs the compliance regarding the common business rules was essential. Like with use cases, I always updated or created the business rules first before assigning changes to RDC. Also, they provided a mean to identify the required test cases.

(1091 of 5000 characters)

Library management tools (AI10)

Used library management tools for source code control and build management.

Provide two instances where you used library management tools. In each case, please briefly describe the circumstances, your role and the deliverables you created.

Guidance for this capability statement

Examples of library management tools are Rational ClearCase, Rational Team Concert, CMVC, SCLM, Librarian, SVN and PVCS. This capability focuses on source code control (CVS, RTC, PVCS) and automated build tools (Make files, ANT scripts, RTC). Advanced topics include forks and branches, level builds, baselines etc.

Example 1: VISTA PV & IK - ClearCase for source code and document management

Duration:* From 01 Mar 2009 To 30 Sep 2013

Project or major activity:*

VISTA PV & IK - ClearCase for source code and document management

Description:*

In VISTA all work products and source code files were maintained by Rational ClearCase (CC) Together with Rational Software Architect (RSA) I used it on a daily base:

- Create / update work products (see AI11)
- Update VISTA code management XML files and run a verification build with an ANT script
- Retrieve the latest code (or a specific branches) of PV or IK for code analysis
- Check-in/out source code for BI development
- Create new baselines and rebase activities (in VISTA 2 main streams with sub streams were used in parallel, code changes needed to be synched among these streams accordingly)
- Create / update testcases for Rational Manual Tester (RMT)

(645 of 5000 characters)

Example 2: SPECTIVE - Microsoft Visual SourceSafe

Duration:* From 01 Aug 2007 To 01 Mar 2008

Project or major activity:*

SPECTIVE - Microsoft Visual SourceSafe

Description:*

In my role as Java developer in the SPECTIVE project (during my assignment in India) the library management tool to be used was Microsoft Visual SourceSafe (VSS). Together with Eclipse as development tool I used it intensively for my development activities:

- Check-in/out source code files
- Run ANT build scripts for creating a local build
- Synchronize updates on one file from multiple developers
- Maintain work products (use cases etc.)

(430 of 5000 characters)

Design from requirements (AI9)

Elaborated and translated functional and non-functional requirements into a design.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the circumstances, your role and the deliverables you created.

Example 1: VISTA PV - Notification Design

Duration:* From 01 Jul 2008 To 01 May 2010

Project or major activity:* VISTA PV - Notification Design

Description:*

Among many other changes, the new PV notification concept is probably the best example in PV where I created a new design from functional and non-functional requirements.

In a first step I gathered the requirements from the lead architects of the business applications in order to retrieve a map of all PV attributes which are relevant to be notified for their business. These requirements were refined later when the applications implemented their notification part. From the customer I got some non-functional requirements in terms of batch run times.

As architect I transformed all these requirements into a complete design, summarized in a technical (in English) and a business (in German) conceptual overview documents. These documents contained state transition tables, interface definitions, configuration parameters and sequence diagrams.

The new design could be implemented and set live successfully.

(907 of 5000 characters)

Example 2: VISTA PV - Optimize PV Search

Duration:* From 01 Jun 2011 To 01 Sep 2011

Project or major activity:* VISTA PV - Optimize PV Search

Description:*

An example where mostly non-functional requirements were involved was the change to optimize the PV search performance.

The PV search is probably the most central functionality in VISTA and is also used out of other business applications. The users however could enter every possible search combination which at the end created a very non performing SQL query, sometimes affecting the entire system due to "long runs" on the DB. In a workshop I proposed to prevent some time-taking search combinations in the GUI. Additionally, I could set field prioritization: if the user entered a social security number (which is unique), then all other attributes would be ignored, leading to a simple search SQL.

Before I gave the search design for implementation, I documented the approach and the specific prioritization levels for the fields in a common understanding document which was later the basis for acceptance testing.

Additionally, I let the DB team analyze the existing indexes on the PV tables. We were able to remove some unused indexes and introduce new ones for frequent parameter combinations.

The PV search performance is now much better and no "hangs" have been recorded since then.

(1187 of 5000 characters)

Methods (AI8)

Used two or more major application development methods.

Provide two instances that demonstrate your use of two or more major application development methods (e.g. LAD / Waterfall, RAD, DSDM, RUP). In each case, please briefly describe the project, your role and the development method you employed.

Guidance for this capability statement

Examples of application development methods are Waterfall, RAD (Rapid Application Development), DSDM (Dynamic System Development Method), RUP (Rational Unified Process) and Agile.

Example 1: VISTA PV - VSDL / Custom Application Development

Duration:* From 01 Mar 2008 To 31 Aug 2009

Project or major activity:* VISTA PV - VSDL / Custom Application Development

Description:*

Custom Application Development is the former AD 2.0 method which was used (among other processes) from the UMF framework to define the VISTA System Delivery Lifecycle (VSDL). This process was engineered to satisfy the special needs of GD and was heavily used during the whole delta development in VISTA. It describes the work products to be created and the responsibilities of each project-member in the various development stages.

I understood the components from this method and was able to efficiently work together with RDC and the customer using the established assets and tools.

(583 of 5000 characters)

Example 2: VISTA PV & IK - Agile Development

Duration:* From 01 Sep 2009 To 30 Sep 2013

Project or major activity:* VISTA PV & IK - Agile Development

Description:*

During and after the scope refinement phase we applied Agile software development method. In this phase I tried to fulfill all of the Agile Manifesto. In brackets [] I mention the concrete situations and actions in VISTA.

- Customer satisfaction by rapid delivery of useful software
[Testcenter offered early and extensive testing sessions with end users]
- Welcome changing requirements, even late in development
[Planning of changes according to their priority and not submission date]
- Working software is delivered frequently (weeks rather than months)
[New releases every two weeks]
- Working software is the principal measure of progress
[Progress measured by the implementation of changes and defects, not budget or time]

- Sustainable development, able to maintain a constant pace
[Many test phases like AIT, SIT, Change Testing]
- Close, daily cooperation between business people and developers
[Architects, testers and product managers in the same room (test center)]
- Face-to-face conversation is the best form of communication (co-location)
[Informal meetings with exact the required persons to discuss specific topics]
- Projects are built around motivated individuals, who should be trusted
[Most of the team members having experience of several years and the will to reach the Go-Live targets]
- Continuous attention to technical excellence and good design
[Bi-weekly meeting with all architects (Central Architecture Board)]
- Simplicity - the art of maximizing the amount of work not done - is essential
[Focusing on important changes]
- Self-organizing teams
[Application leader are also SMEs, architects act as stand-ins for application leader]
- Regular adaptation to changing circumstances
[Constant replanning of changes and defects, taking over additional responsibilities]

(1750 of 5000 characters)

Configuration management tools (AI7)

Used configuration management tools for requirements, change requests, defects or traceability.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the circumstances, your role and the deliverables you created.

Guidance for this capability statement

Examples of configuration management tools are Clearcase, Clearquest, Rational Team Concert and CMVC.

Example 1: SPECTIVE - FocalPoint

Duration:* From 01 Aug 2007 To 01 Mar 2008

Project or major activity:* SPECTIVE - FocalPoint

Description:*

In the SPECTIVE project I used Telelogic FocalPoint, a system for management of product and project portfolios. In this project it was used to assign changes and defects to a developer, tester or back to the customer. I had to estimate the tasks which were described in the item and - when the effort was approved - start the implementation. I also updated the status and spent effort at least weekly, as it was the main tool for tracking the overall project progress.

(468 of 5000 characters)

Example 2: VISTA PV & IK - Rational Clearquest

Duration:* From 01 Mar 2008 To 30 Sep 2013

Project or major activity:* VISTA PV & IK - Rational Clearquest

Description:*

In VISTA I daily worked with Rational ClearQuest (CQ) as the main tool for managing defects and changes, but compared to the SPECTIVE project, in a much activer role. For both PV and IK I was responsible to assign defects and changes without delays to the correct team member. And I had to track the progress of each item myself, especially in the role as Application Leader. As each item also got a sub release assigned, I needed to supervise them all regularly and take the appropriate actions if a release could not be achieved.

ClearQuest information had to be kept in synch with other management tools like SharePoint and TrueAct (for incident management). With TrueAct the end users could submit incidents from production for analysis to IBM or other stakeholders. Many times the analysis of an incident resulted in new AMS defects or new changes. In both tools I needed to put a cross-reference: the service desk needed to know when a defect would be implemented and the ticket owner would get his problem completely solved. The TrueAct reference in CQ enabled me to get additional information to a defect, respectively a "live" example from production.

(1159 of 5000 characters)

Complete solution (AI6)

Designed, built, tested and packaged a significant aspect of a complete solution to meet the requirements of the client.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the solution, your role and the deliverables you created.

Guidance for this capability statement

Choose projects in which you are involved from end-to-end; i.e. from solution outline to testing/deployment. Focus on describing your personal contributions at each stage.

Example 1: VISTA PV - Integration of external application "BB"

Duration:* From 01 Jan 2012 To 01 Mar 2012

Project or major activity:* VISTA PV - Integration of external application "BB"

Description:*

The application BB (blind expenses / Blindenbeihilfe) was implemented as a small web-application outside of VISTA but uses VISTA PV data. In particular, BB does have to administer correspondence- and payment-addressees within PV. The application gets its data from PV through a DB-synchronization where everything of importance to BB is read from the VISTA-DB and transferred to the BB-DB. I had to make sure that BB can administer the relevant data within VISTA PV.

First I needed to outline the solution in a concept. Within an ARC 118 change case document I described the initial situation, the technical components to be changed (CM, GUI adaption, a new ELAR WorkPerformer, new use cases and business rules and new privileges).

Secondly, I defined the required tasks and responsibilities for all component changes which involved activities from the architect, developers and testers. The sequence of these activities were also aligned by using several features and a tracking sheet. I had to start first with my activities due to some technical dependencies. The RDC developers started with the next sub release, but the testers could already start with defining test scenarios for this change.

Before the developers checked in the code for the next build, I held a spot check session with them and the tester to verify the most important points are working. When the developers executed the AITs and re-assigned the change to me, I could wait for the official build and meet with the team leader from the client. Together we executed the SITs and change test cases from the customer. As no problems occurred and all test cases passed successfully, the new functionality could be delivered for production.

(1707 of 5000 characters)

Example 2: VISTA IK - Create solutions for interim releases (iFix)

Duration:* From 01 Oct 2010 To 30 Sep 2013

Project or major activity:* VISTA IK - Create solutions for interim releases (iFix)

Description:*

With go-live of phase E1a I had to deal with incidents from production. These incidents were the trigger for analyzing the root cause of the customer's problem. An AMS incident could lead to code changes, either as defects or Changes.

Especially in IK with its exposure to other SVAs it sometimes happened that a Defect was too severe to have it implemented with the next official release (which could be delivered up to three months later). In such cases, an emergency interim build (iFix) was required. As this implied high efforts on all sides (development, infrastructure, build team, service provider and test management), these iFixes needed to be negotiated with all parties. The delivery manager from IGS however first needed an impact analysis on business and system level.

When all parties agreed on a specific delivery date, I had to assign the defect quickly to RDC. Here, a lot more tracking and interaction was required compared to normal defects. I immediately worked on the definition of test cases which would be required for acceptance. Other than for normal defects, an iFix activity has to work on the first try as we would directly patch the productive system. Therefore I often also held a handover call as we did for complex changes to assure a common understanding. Before RDC marked the iFix defect as "implemented", we verified the solution in a spot check session, for very critical defects also together with the customer.

Finally, when all tests passed successfully, we could deliver the patch. But I also had to make sure that the synchronization to all development streams would be adhered. For this I created synchronization defects and assigned them to RDC to assure the fix would also work in the upcoming regular build.

(1750 of 5000 characters)

Development progress and quality tracking (AI5)

Defined and measured metrics tracking development progress and quality.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the circumstances, your role and the deliverables you created.

Guidance for this capability statement

Example metrics that can be used to track progress include: defect statistics (creation rate, closed rate, avg cycle time), team velocity, # of use cases implemented, etc. Code quality metrics such as cyclomatic complexity, Maintainability Index, unit test code coverage can also be used as example here. Document how you adopted the use of these metrics and their benefits and challenges (if any.)

Example 1: VISTA PV - Test Automation Tracking

Duration:* From 01 Aug 2009 To 01 Mar 2011

Project or major activity:* VISTA PV - Test Automation Tracking

Description:*

In the test automation sub project I was responsible for a constant progress on automated test cases for PV. Before, during and after a special workshop in India I tracked the progress with several tools and metrics.

One tracking sheet listed all test cases from Application Integration Testing (AIT) in PV. In a first phase I marked the automation candidates with a flag in the sheet, respectively I organized an Update in Rational ClearQuest Test Manger (CQTM) on a special field to indicate the state of an automation candidate (like Auto_Planned, Auto_Ready, Auto_Implemented). When a new AIT was automated or ready for automation, the team updated the status in CQTM accordingly; the tracking sheet could be updated on button-click for all items at once. With this I could already show a rough progress state.

For the 3-weeks-workshop PV had separate targets (50 new automated AITs in 15 days). In order to track this target I counted the new AITs each day and forwarded the results to the VISTA test manager, who made a separate tracking sheet for the workshop - and I could communicate to my application leader that we have reached the target after 9 days already.

The overall progress tracking needed to be refined however, as there were short and very complex AITs involved. Therefore I introduced a complexity factor, based on the number of steps in the AIT. This provided a much clearer picture and allowed a better planning. Additionally, in order to consider AITs with small defects or AITs which were not even translated I set a completion factor for each AIT (e.g. 90% if a small step was failing). All these measurements led to a very accurate automation progress tracking.

(1686 of 5000 characters)

Example 2: VISTA PV & IK - Change Project Progress Tracking

Duration:* From 01 Aug 2011 To 01 Jan 2013

Project or major activity:* VISTA PV & IK - Change Project Progress Tracking

Description:*

In my role as application leader for PV (and stand-in for IK) I tracked the development progress of these applications and reported it in a weekly status meeting to the management. The progress was mainly measured by implemented changes and the completion rate of all changes, respectively. To track this metric I used a tracking sheet which retrieved data from ClearQuest (CQ). From the fields "current / estimated effort on analysis, implementation and testing" I summarized the information to simpler statements like "16 person days open for release SR14".

One of the difficulties lied in the mapping between releases and quarter years: the "dates" of

a sub release could hardly be matched to fixed calendar dates, because a release might be set productive at a specific date like March 31, but from IBM side the changes of this release were "completed" almost one month before (in the remaining time the customer executed their tests and build activities started). This fact was important to consider, as the customer ordered an amount of effort to spend per quarter and not release. Therefore I needed to calculate very carefully which changes could be implemented when, how many defects we could afford and how the entire team can work on billable items - considering the capacity and availability of each team member.

All these data had to be simplified every week, but it helped a lot managing the applications: I could immediately answer customer's questions like "Can we order an additional change in Q2?" or "Which changes should be taken out of scope for the next release, if we increase the team's capacity by 5 person days and order the big change XY instead?".

(1673 of 5000 characters)

Automated development tools (AI4)

Used automated development tools.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the circumstances, the tools, your role and the deliverables you created.

Example 1: VISTA PV - RSA for BI Development

Duration:* From 01 Sep 2008 To 01 May 2009

Project or major activity:* VISTA PV - RSA for BI Development

Description:*

The BackInterface (BI) was developed and maintained in Switzerland and not in India. I implemented several Defects and Changes for BI with IBM Rational Software Architect (RSA).

Using the modeling perspectives and the "Vista transformation plugin" for RSA I could generate components (following the model driven approach) and its code in a consistent way, avoiding to code hundreds of lines just to have the default headers and methods created.

Using Apache Maven I had a standardized way to configure and build the application in an easy way, assuring all required projects and plugins were loaded in the correct version.

Another RSA plugin I used in VISTA was Code-Style-Checker which served the uniformity of the code and reduced implementation errors.

(751 of 5000 characters)

Example 2: Android App - Android Development Tools (ADT)

Duration:* From 01 Aug 2012 To 01 Nov 2012

Project or major activity:* Android App - Android Development Tools (ADT)

Description:*

As part of my "Giveback" in 2012 I planned and implemented "Bluepages Contact Sync", an Android App which allowed to take over contact details including profile pictures from IBM Bluepages to the phone's contacts.

For this project I used Googles Android Software Development Kit (SDK), consisting of a debugger, libraries, an Emulator, documentation, sample code and tutorials. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin.

Using these the functionality of this toolset allowed me to extend a sample project, apply different configurations and build, upload the App to the Emulator and run the debugger for it on mouse-click.

(704 of 5000 characters)

Test and debug (AI3)

Performed unit test and debugged complex software as defined in a test plan.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the unit under test, the test strategy, test process and your role.

Example 1: VISTA PV - JUNITs for BI Development

Duration:* From 01 Aug 2008 To 01 May 2009

Project or major activity:* VISTA PV - JUNITs for BI Development

Description:*

For the BackInterface (BI), an entire test suite had been created over time. When I implemented changes and defects for BI I extended and updated this suite with new JUNIT test cases and assured a high test coverage of the BI code.

The test suite allowed to execute an entire regression, build verification or combinations of a logical sequence of JUNITs. I could run and supervise the program flow and the data processing of new or modified methods and verify that the rest of the program would be unaffected.

For analyzing incidents from production I just had to modify the test data and run some JUNITs in debug mode.

(618 of 5000 characters)

Example 2: Android App - Debug software for mobile devices

Duration:* From 01 Aug 2012 To 01 Nov 2012

Project or major activity:* Android App - Debug software for mobile devices

Description:*

My own Android app "Bluepages Contact Sync" (See AI4 & [Giveback]) was developed with the Android Development Tools (ADT) for Eclipse. Testing the application confronted me with several challenges:

- The Emulator could not connect with the IBM Intranet
- Android exists in many versions and phone producers tend to customize the pure Android system, leading to new features but also some incompatibilities
- I had "only" two different physical Android devices for testing the app in "real world"

To circumvent the intranet access problem I downloaded the HTML pages of several different

Bluepages search results, anonymized the data and uploaded the files on a private web server. In the app I defined two "profiles" which I could change on the fly. In the "test"-profile the app just used different URLs to my web server instead of the real Bluepages.

The problem with the different Android versions could be mitigated a bit with the ADT, offering to run the app in many different Android versions. The biggest difficulties I had to overcome was that a core component (the contact framework) changed significantly between Android 2.x and 4.x. Plus, the contact app on the devices are very often customized.

Using log messages (which could be turned on and off depending on the profile) and an app to send the phone's log files to a computer (SendLog) and having the app installed on two different real devices I had everything in my hands to analyze bugs which occurred on one device only with real data from Bluepages.

(1508 of 5000 characters)

Programming language selection (AI2)

Compared the possibilities, strengths and weaknesses of two or more programming languages to make recommendations for a business and technical context.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the circumstances, your recommendation and rationale.

Guidance for this capability statement

There are many factors for comparison. Interpreted languages are often quicker to develop, but run slower than compiled languages. For some languages, there are good toolkits available for GUI development, web services etc., for others you have to code everything yourself. Also consider the types of languages - functional, object-oriented, logical, imperative etc. Some languages are platform-independent, others are tied to a specific Operating System and hardware. Don't forget the business aspect. Maybe Fortran isn't the best technical option, but if 90% of the programming staff is familiar with it and this is an emergency project, it may be the best overall solution. Also, some companies are uncomfortable locking themselves into a proprietary language like C#.

Example 1: VISTA PV - DB2 Stored Procedure

Duration:* From 01 Jun 2011 To 08 Aug 2013

Project or major activity:* VISTA PV - DB2 Stored Procedure

Description:*

When I had to create complex data corrections and mass updates which involved business logic, I usually had several options to choose a language. One would have been to use a new or modified batch, write the logic in Java and integrate it into VISTA. Benefits would have been to use existing services and business rules checks, but having the disadvantage of dependencies to the build plan and additional effort e.g. for the batch team.

Therefore I tend to use DB2 functionality: DB2 expands on the definition of a stored procedure by allowing you to code them in just about any language you may need. DB2's external procedures provide an option to implement more complex logic than SQL can support, using languages such as Visual Basic .NET (VB.NET), C, C++, C#, COBOL or Java.

So again, I could choose which language to use, often it was a decision between SQL and

Java. Normally, with SQL no control flows are possible, which would be a clear argument for JAVA. But DB2 offers to use conditional and loop statements (if-then-else, while, for, ...) as in other programming languages. Arguments for JAVA would have been a full-featured exception handling model and object-oriented principles. But for my purposes this was not a requirement, and because a stored procedure in JAVA requires a compilation task and SQL does not, my choice for SQL (making use the conditional statements) was justified.

(1397 of 5000 characters)

Example 2: Android App - Java vs. C++

Duration:* From 01 Aug 2012 To 01 Nov 2012

Project or major activity:* Android App - Java vs. C++

Description:

When I dived into Android programming and started on my Giveback-project "Bluepages Contact Sync" (see AI2 and [Giveback]) I had to choose among many environments, tools and languages:

- Android Software Development Kit (SDK):
The "official" development kit based on Eclipse using JAVA)
- Native Development Kit (NDK):
C and C++
- App Inventor:
Web-based visual development environment for novice programmers
- HyperNext Android Creator (HAC):
HyperNext is an interpreted English-like scripting language
- The Simple project:
Simple is a BASIC dialect for developing Android applications
- Qt for Android:
C++ and QML
- PhoneGap
Use HTML and JavaScript code to function like a mobile app
- ...

Out of these tools and programming languages, I eliminated PhoneGap quickly. Even though the compatibility (HTML and JavaScript) and ease-to-use were arguments, I had to deal with internal phone functions which were not accessible by this combination. The NDK would have allowed to create high-performing native applications in C, C++, but I knew my app wouldn't become very CPU heavy. Also the tool setup seemed to be more complex. QT for Android using C++ and/or QML was interesting, as it offered good portability features (for example for iOS). But, at that time, the available documentation, sample code, libraries and up-to-date feature list for the official SDK for Android using JAVA was the main motivation to start with the Android SDK and not with QT. Additionally, my skills in JAVA are better and required less introduction effort.

(1506 of 5000 characters)

Application programming (AI1)

Programmed in one mainstream programming language, according to project guidelines and coding standards.

Provide two instances that demonstrate you have met this criterion. In each case, please briefly describe the circumstances, the purpose of the code you wrote and whether your code was successfully deployed. Please also identify the applicable project guidelines and coding standards.

Guidance for this capability statement

A mainstream language is something like Java, C++, PHP, C# etc. It's important that you show some sort of structure here. For something like Java, you could refer to the Sun Java coding standards. Or perhaps you developed a standard for use within your project for cascading Makefiles.

Example 1: VISTA PV - BI Development in Java

Duration:* From 01 Aug 2008 To 01 Aug 2009

Project or major activity:* VISTA PV - BI Development in Java

Description:*

While most of the development work in VISTA had been outsourced to RDC India, the Back Interface (BI) remained to be developed and maintained in Switzerland for various reasons. I implemented several changes and defects in the VISTA framework (which is based on JAVA Spring / Hibernate). For modeling changes I used the VISTA Transformation plugin (see AI4). For updating the unit tests, I programmed new JUNITs for the existing test suite.

I adhered the VISTA coding standards, which are based on the "Oracle Java Coding Standards" and could be verified with Checkstyle, a plugin for RSA. I could further improve the code quality by using a Programming Mistake Detector (PMD), which is a static ruleset based Java source code analyzer that identifies potential problems.

(770 of 5000 characters)

Example 2: BoomCMS - Development in PHP

Duration:* From 01 Jan 2010 To 30 Sep 2013

Project or major activity:* BoomCMS - Development in PHP

Description:*

In my spare time I support the website for a local community. Initially, I programmed the entire website in ColdFusion Markup Language (CFML), HTML, CSS and Access databases. As the site needs to be migrated to another host which does not support CFML but only PHP, I started to develop an entire Content Management System (CMS) in PHP. The requirements from the community to this website were very specific and could not be covered with an existing open source solution.

In order not to reinvent the wheel again I decided to use a framework which supported many basic features like forms, templates, security, session handling and more. My choice was the PEAR framework for PHP, which is open source and served my needs perfectly. I followed the existing coding standards (<http://pear.php.net/manual/en/standards.php>) in case other users want to use or modify my CMS: it is available in a beta-version on SourceForge

(<http://sourceforge.net/projects/boomcms/>).

(960 of 5000 characters)

Development

Provide examples of career development activities.

In the Education section, list all educational activities that support your skill growth. You are expected to average 40 hours a year.

*Technical focus and specialty areas (average 30 hours per year)

If you have educational activities that were not found in your Employee Learning History, please add them in the additional documentation section.

Learning activity	Activity code	Duration	Status	Date
Quickstart for TeamSD	WBT7989B	1.5 Hours	Completed - Successfully	06 Sep 2013
Android Programming	ADHOC- 000000000688346	80.0 Hours	Completed - Successfully	31 Oct 2012
Introduction to Programming for Mobile Applications - Developing an Application for Android	SSEK3323	60.0 Minutes	Completed - Successfully	18 Jul 2012
Automated migration of HOST-based legacy applications using ACT4 [8Q7H9J]	ADHOC- 000000000908561	2.0 Hours	Completed - Successfully	02 Mar 2012
DB2 Stored Procedures Programming	ADHOC- 000000000688061	24.0 Hours	Completed - Successfully	08 Jul 2011
HackDay8	ADHOC- 000000000688759	8.0 Hours	Completed - Successfully	22 Jul 2010
Lotus Notes Plugin Programming	ADHOC- 000000000688317	40.0 Hours	Completed - Successfully	09 Apr 2010
Architectural Thinking	SIMT2G	27.0 Hours	Completed - Successfully	22 Jun 2009
IBM India Onboarding Training Program	ADHOC- 000000000689836	240.0 Hours	Completed - Successfully	05 Aug 2007
IBM Unified Method Framework for TI Professionals Workshop	TIWSBX	24.0 Hours	Completed - Successfully	06 Jun 2007
Introduction to SOA and Web Services	BLG0328	2.0 Hours	Completed - Successfully	26 May 2007
IBM Global Services Method for Technology Integration Professionals	TICAD01	24.0 Hours	Completed - Successfully	26 May 2007

Learning activity	Activity code	Duration	Status	Date
UMF Practical Use Case Modelling	A4976	24.0 Hours	Completed - Successfully	24 May 2007
SAP Methods & Tools eLearning	BLG0349	4.0 Hours	Completed - Successfully	19 May 2007
Component Modeling	A5064	16.0 Hours	Completed - Successfully	09 May 2007
Principles of Use Case Modeling with UML 2.0	DEV111V2	2.0 Hours	Completed - Successfully	05 May 2007
ARC-110 - Architectural Thinking for On Demand Business	LTU8210F	2.0 Hours	Completed - Successfully	05 May 2007
Principles of Modeling	DEV110	8.0 Hours	Completed - Successfully	28 Apr 2007
Essentials of IBM Rational Application Developer: Getting Started	DEV335	2.0 Hours	Completed - Successfully	28 Apr 2007
Global Services Method Fundamentals	GSMF100B	24.0 Hours	Completed - Successfully	14 Apr 2007

*Teaming, Leadership or Business (average 10 hours per year)

If you have educational activities that were not found in your Employee Learning History, please add them in the additional documentation section.

Learning activity	Activity code	Duration	Status	Date
AMS Face to Face and Virtual Workshop	GBS1570	12.5 Hours	Completed - Successfully	17 Jul 2013
GBS Big Data - Using Big Data to Transform an Enterprise course	LT-1227	60.0 Minutes	Completed - Successfully	24 May 2013
Corporate Service Corps Application 2013	ADHOC-000000000689636	32.0 Hours	Completed - Successfully	03 May 2013
Our Path Forward: Front Office Transformation	LT-909	2.0 Hours	Completed - Successfully	02 Apr 2013
Lean Six Sigma - Incident Management Process Workshops	ADHOC-000000000908950	16.0 Hours	Completed - Successfully	21 May 2012
The Right Strategy for Your Way into the Clouds [8Q7HAM]	ADHOC-000000000908680	2.0 Hours	Completed - Successfully	09 Mar 2012
Implementation of Business Rules - stumbling block in migration	ADHOC-000000000908435	2.0 Hours	Completed - Successfully	22 Feb 2012

Learning activity	Activity code	Duration	Status	Date
projects 8Q7H7T				
Lessons Learnt from Application Development Projects [8Q7GYT]	ADHOC-000000000908244	2.0 Hours	Completed - Successfully	10 Feb 2012
Lean Concepts	OPER01A1	16.0 Hours	Completed - Successfully	04 Nov 2011
Client Focus	SGC50940	2.0 Hours	Completed - Successfully	05 Aug 2011
Complex SI Estimating Module 1 - CSI Estimating Framework	BLG2313	2.0 Hours	Completed - Successfully	08 Apr 2011
Performance Engineering: Delivering Successful Projects	PEDSP00B	22.0 Hours	Completed - Successfully	18 Jan 2010
Project Management Fundamentals	PM10G	24.0 Hours	Completed - Successfully	08 Jul 2008
Fascination Communication Success	3PFKOAIC	24.0 Hours	Completed - Successfully	28 May 2008
Global Delivery: Intercultural Workshop	3PICWACH	20.0 Hours	Completed - Successfully	31 May 2007
Project Management Orientation	PM54G	32.0 Hours	Completed - Successfully	14 Apr 2007

Additional learning activities

The preferred method for adding additional learning is to enter it into your [Talent@IBM](#) profile as described above, but if you are unable to enter or import learning from your Talent@IBM profile, or if you would like to supplement what is imported, you may add it here.

(0 of 5000 characters)

Professional giveback

Please specify your giveback activities in at least two (2) of the categories listed below. You must include at least one example for the mentoring category.

Example 1

Duration:* From 01 Sep 2012 To 31 Oct 2012

Number of hours:* 80

Giveback category:*

ERROR: syntaxerror
OFFENDING COMMAND: --nostringval--

STACK: