



THINKTANK CONSULTING
PROJECTS. PROCESS. OPTIMIZATION.

Executive Summary

- How to integrate good Project Management and Business Analysis practices so that the whole process is:
 - Cheap
 - Quick
 - User-friendly and
 - Efficient?
- Solution:
 - Make sure PM and BA documents are interconnected.
 - Assign each Issue, Feature, Use Case and Problem Report a unique ID to track and bind them together.
 - Get developers involved as early as possible (i.e. Planning Stage).
 - Conduct stakeholder meetings every time the new PM or BA document is finalized.
 - Treat all PM and BA documents as live organisms that evolve with time.

Glossary

- **PM** – Project Manager
- **BA (BSA)** – Business Analyst (Business Systems Analyst)
- **Dev** – Developers
- **Test** – Testers
- **SRS** – Software Requirements Specifications document
- **WBS** – Work Breakdown Structure
- **UC** – Use Case
- **PR** – Problem Report

Integrating Project Management and Business Analysis

By Jamal Moustafaev, BBA, MBA, PMP
President & Principal Consultant
Thinktank Consulting

Introduction

One of the projects I've been involved in recently has faced an interesting predicament. Although we knew about good project management and proper business analysis practices, one question remained open: how do we integrate these two methodologies into one seamless process?

We knew that there were some software packages out in the market that provided at least a partial solution to our problems. However, due to the multitude of these products we did not have enough time to choose appropriate software. Furthermore, the products that appeared to be good had a very high price tag. So, the first challenge we had was to come up with an efficient methodology to integrate our project management and business analysis processes and the second challenge was that we were pretty much limited to *MS Office Professional Edition*.

Constraints

Based on the above mentioned information, the objectives and constraints of our endeavor were:

- Cheap (no additional budget was allocated for extra software purchases)
- Quick (we didn't have a lot of time before the actual release of the product)
- User-friendly (developers, testers and especially stakeholders were not willing to spend a lot of time learning the new system)
- Efficient (it had to do the job, of course!)

Solution

• **The Components**

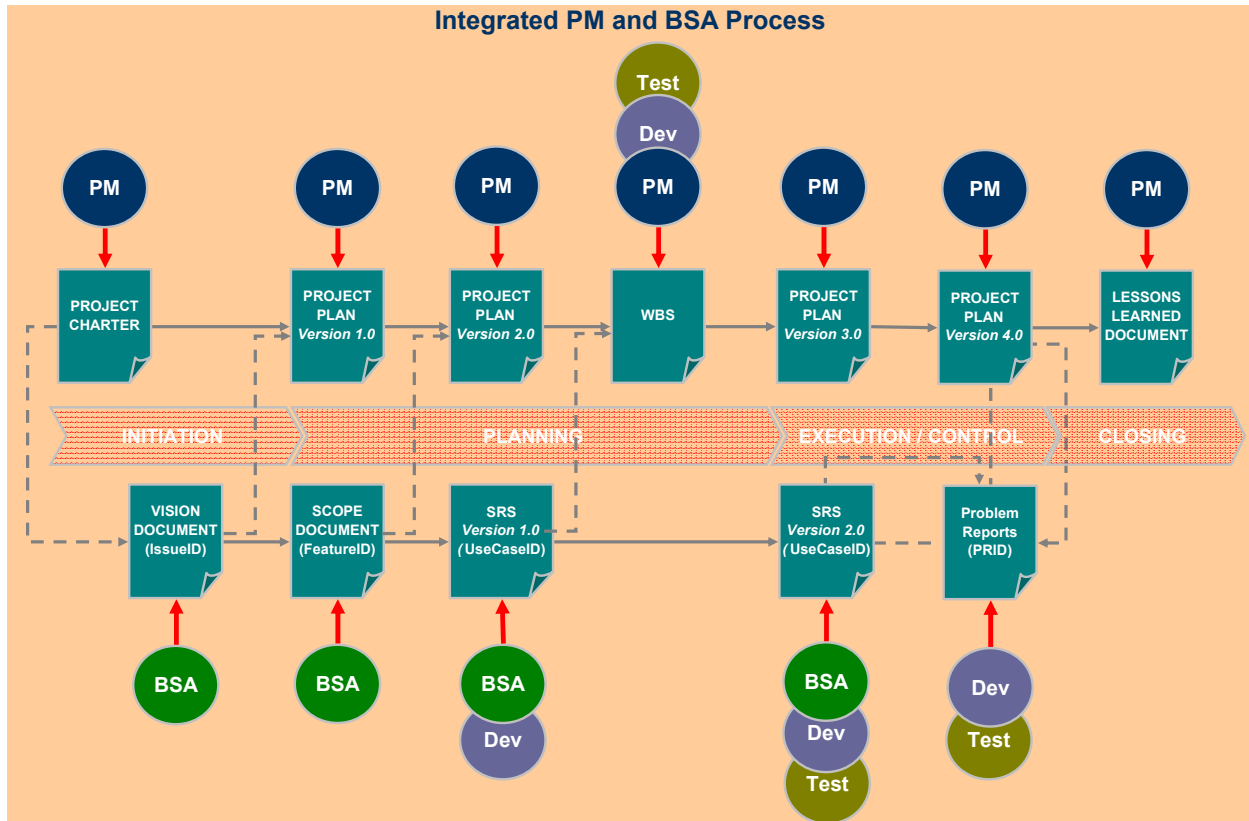
The solution we came up with had the following components:

- Project Management and Business Analysis documents frequently have a lot in common, so they have to feed off each other.
- Each issue, feature, use case and problem report should be assigned a unique ID to track and bind them to one another.
- Developers should get involved early in the process to avoid "technical" surprises in the future.
- Completion of every significant document should be followed by a stakeholder meeting in order to minimize the gap between the expectations and the actual product.
- All project management and business analysis documents should be treated as live organisms that constantly change, mutate and, hopefully, improve.

(Note: I will use an example of developing a software program for ATM from now on in order to better illustrate the processes and to liven up my boring narrative).

So, let's assume that we have been commissioned to build a new software package for an ATM. Let's also assume that the old software for the ATM was deficient and generated a lot of customer complaints. We would, of course, start our project at the Project Charter stage (see *Figure 1: Integrated PM and BSA Process*).

Figure 1



Initiation Stage

- Project Manager completes the Project Charter.
- Business Analysts creates a Vision document.
- Vision document lists all Issues (problems) with previous releases of the software.
- Each Issue in the Vision document has:
 - Name
 - Unique ID (IssueID)
 - Description
 - Who it affects
 - Impact description

• The Initiation Stage

Once the Project Charter is completed by the Project Manager, Business Analyst will start meeting with stakeholders and compile the list of Issues or Problems with the current software. This list becomes an integral part of the Vision document. A couple of sample entries into the Vision document may look like this:

Issue Name	PIN Recognition
IssueID	0001
Issue Description	Current ATM software does not have the capability to accept and validate customer's secret PIN number (Yes, it is THAT bad!©)
Affects Impact	Customers High security risk

Issue Name	Access to All Accounts
IssueID	0002
Issue Description	Current ATM software loads Checking accounts only, ignoring Savings, Credit Card and Mortgage accounts
Affects Impact	Customers Customers can not access all of their accounts, thus leading to inefficiencies and low customer satisfaction

Planning Stage

- Business Analyst creates a Scope document.
- Scope document contains a list of Features requested in the new software package.
- Each Feature in the Scope document has:
 - Name
 - Unique ID (FeatureID)
 - Description
 - Benefit
 - Issue(s) it addresses
- Stakeholder meetings are held after Vision and Scope documents are finalized.
- Both Vision and Scope documents are published.
- Project Manager can update and improve the Project Plan after publication of each document.
- Business Analyst and Developers create a Software Requirements Specification document.
- SRS document contains Use Cases with step-by-step descriptions of the interactions between the User and the System.
- Each Use Case in the SRS document has (among other things):
 - Name
 - Unique ID (UseCaseID)
 - Description
 - Feature(s) it addresses.
- Stakeholder meetings are held after SRS documents is "finalized".
- SRS document is published.
- Project Manager can now develop a WBS and determine Work Packages using using Use Cases in the SRS document.

Note that each Issue has a unique IssueID assigned to it. The purpose of this ID is twofold: it will be used both for tracking and reporting purposes.

• The Planning Stage

Once the Vision document is complete, the BA moves on to the Scope document. The Scope document will list (among other things) all high-level Features that address Issues outlined in the Vision document. Let's look at several sample entries in the Scope document:

Feature Name	PIN Number
FeatureID	0001
Feature Description	New ATM software will be able to prompt for, accept and validate customer's secret PIN number
Benefit	Improved security
Priority	Must Have
Addresses (Issue)	0001

Feature Name	Accounts Menu
FeatureID	0002
Feature Description	New ATM software will allow the customer to access his/her Checking, Savings and Credit Card accounts
Benefit	Improved customer service
Priority	Must Have
Addresses (Issue)	0002

Feature Name	Accounts Menu – Mortgage
FeatureID	0003
Feature Description	New ATM software will allow the customer to access his/her Mortgage account
Benefit	Improved customer service
Priority	Should Have
Addresses (Issue)	0002

Note, that in the above examples we have "one-to-many" relationships between Issues and Features. In real life, however, these relationships can be "one-to-one", "many-to-one" and, even, "many-to-many" if required. Also, our examples show Issues and Features tables as they appear in MS Word; however nothing prevents a Business Analyst in any given project from entering them into a spreadsheet or even some kind of database (e.g. MS Access).

At completion of both Vision and Scope documents these files are published and made available to the Project Manager, who can use them to yet again update and improve the Project Plan (see *Figure 1: Integrated PM and BSA Process*).

Once the Scope document is complete, it is time for Business Analyst and Developers to start working on the Software Requirements Specifications document (SRS). Our experience shows that it is VERY important to have the Developers present in the room when the SRS is being discussed. What happens very often is that BA works on requirements and the SRS alone or with stakeholders and later they discover that technology does not allow for some particular "must-have" feature to be built. This constitutes the dreaded scenario when a PM has to go back to the stakeholders and deliver the bad news that the promised feature is no longer available.

SRS contains Use Cases that, as we all know, describe step by step actions of the User and the System. Here is what a header of a typical Use Case could look like:

Use Case ID:	0001
Features Addressed	0001 and 0002 (see Vision document and Scope document)
Use Case Name:	Accessing ATM Services
Created By:	John Smith
Last Updated By:	Mary Brown

Date Created:	13-Jun-2003	Date Last Updated:	30-Jul-2003
----------------------	-------------	---------------------------	-------------

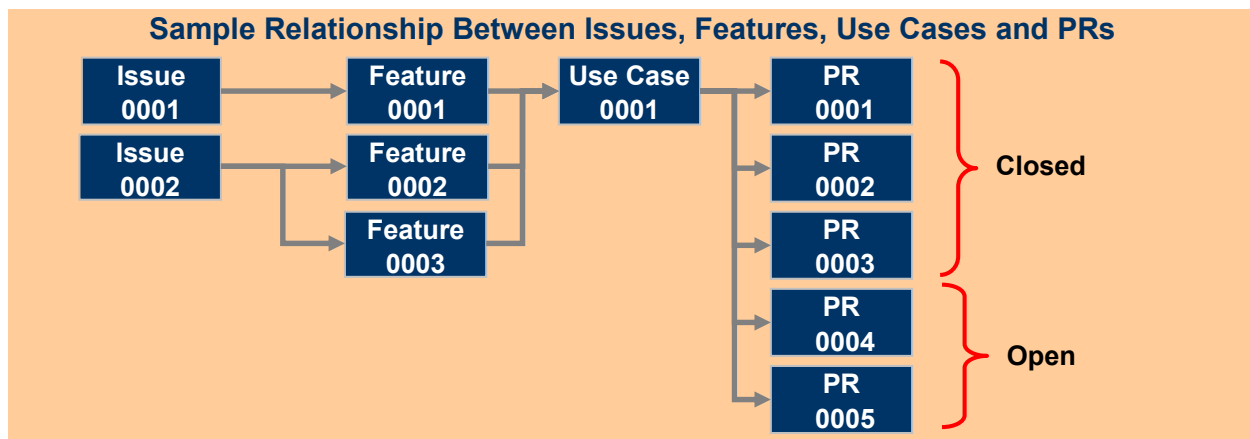
As you can see, each Use Case is assigned a unique ID that ties it to all relevant Features. So, if at this point of time someone asks you a question:

- Have all of our Issues been addressed by the Features? or
- Did we miss any of the Features when developing the Use Cases?

it is very easy to come up with a quick answer by listing all the Issues, Features and Use Cases (see *Figure 2: Sample Relationship Between Issues, Features, Use Cases and PRs*). Again, some bright mind in your organization may decide to utilize some kind of a database tool to list and organize them. Having said that, I just want to point out that Excel does the job too!

Now, that the first draft of the SRS is ready (and, yes, you better believe that there will be many more drafts in next several months!) the Project Manager can utilize this document to come up with the Work Breakdown Structure and break down all the work into “work packages”. At this point of time the Project Plan can be “finalized” to the +/- 10% degree of accuracy with respect to budget, resources and time.

Figure 2



Execution & Control Stages

- Business Analyst and Developers create a Software Requirements Specification document.
- Changes and updates start pouring in.
- PR is generated every time there is a:
 - Change or
 - Technical issue.

• The Execution and Control Stages

Actual development starts In this stage of the project; and if you think that your SRS and Project Plan are picture perfect, you are in for a very nasty surprise! Suddenly the Use Cases and diagrams your team labored on for weeks appear to have missing Alternatives and Exceptions and your technical guys suddenly inform you that what they thought was achievable quickly a couple of weeks ago would actually add an extra 3 to 6 months to the project schedule. And let’s not forget all the “additional” and “overlooked” requests by the stakeholders that increase the scope of the project and have to be filtered through the Change Committee!

So, how does a PM/BA tandem deals with all these changes, updates and revisions? The answer we came up with was Problem Reports. Every time there is a need to update the SRS and the Project Plan a PR is filled out and distributed to all team members. Here is a simplified version of a Problem Report:

PRID	0005
Problem Description	No Exception to handle a wrong PIN
Relates to Use Case	0001
Environment	Staging

Execution & Control Stages (continued)

- Each PR has (among other things):
 - Unique ID (PRID)
 - Description
 - Actions Taken
 - Use Case it relates to
- SRS document is updated by the Business Analyst based on the relevant PRs.
- Project Manager updates the Project Plan based on the relevant PRs.
- Once the PRs are resolved, they are closed by the BA, PM, developers or testers.

Closing Stage

- Project Manager compiles and analyzes all the PRs statistics and other relevant information
- This data can now be :
 - Presented to the stakeholders and the team, and
 - Included in the “Lessons Learned”

Status	Assigned
Priority	Must Have
Change Request?	No
Originator	Tester
Owner	Business Analyst
Date Opened	30-Jul-2003
Date Last Updated	01-Aug-2003
Actions Taken	30-Jul-2003: No Exception to handle a wrong PIN 01-Aug-2003 New Exception (Exception 3) to handle a wrong PIN is now added to SRS (Use Case 1, Exception 3). Please, see the updated version.

Once the PR is resolved, it can be “closed” by its owner (e.g. BA, tester, developer, etc.)

Once again, I would like to point out that it doesn't hurt to have some kind of a database tool to input all this information. But in the worst-case scenario, even an Excel spreadsheet will do the trick.

The first and most obvious benefit of this system is the ability to track the issues as they come along and to assign them to the right people. However, there are also some additional benefits that may be overlooked initially. For example, how many times does a Project Manager has to answer question like:

- How many PRs do we still have open at this point of time?
- How many PRs have been successfully closed?
- How many PRs resulted from “change requests” approved by the Change Committee?
- How many PRs resulted because of the “technical issues”?
- Who is responsible for this particular PR?
- How long has this particular PR been unresolved?
- What is the average “PR resolution time”? etc.

If you have access to the information presented above, the answers to these questions are readily available (especially if the PM doesn't forget to bring his laptop to the meeting ☺)

• The Closing Stage

One of the most important (and most overlooked) parts of the closing stage is the compilation of “Lessons Learned” document. Let's look at *Figure 2: Sample Relationship Between Issues, Features, Use Cases and PRs* one more time. Do you think it would be useful to you, your team members, stakeholders and other PMs in your organization to have the following information and statistics included in the “Lessons Learned” report? Here is a sample:

- The final product includes:
 - 100% of “must have” features,
 - 60% of “should have” features,
 - 10% of “nice to have” features.
- 25% of all PRs resulted from internal technical and programming-related problems.
- 35% of all PRs resulted from incomplete Use Cases in the SRS.
- 40% of all PRs were a direct result of changes requested by the stakeholders.
- 99% of PRs were successfully closed in the course of the project.

- 1% of the PRs was transferred to the support team.
- PRs resulting from changes requested by the stakeholders added extra 2 months of development and \$250,000 to the initial estimates.
- Use Case 0001 (Accessing ATM Services) generated an unusually high number of problem reports (75%) and in our opinion this due to lack of unique Customer ID in our banking database, etc.

I sincerely hope you will agree with me that information like this would be "worth its weight in gold" both in Project Closing meeting with the stakeholders and in the "Lessons Learned" document!

Conclusion

Hopefully what I managed to do in this article is to demonstrate that PMs and BAs do not necessarily need fancy and expensive software packages or complex organizational structures to handle large projects. The process I described can be implemented even at a small and relatively unsophisticated (from a Capability Maturity Model point of view) company. All you have to do is:

- Tag Issues, Features, Use Cases and Problem Reports with unique IDs and track them throughout the project.
- Make sure that PM documentation is updated every time BA documents change (and vice versa).
- Start talking to developers early in the process.
- Do not assume that Project Plan or Software Requirements Specifications will ever be perfect. (Well, at least not until the project is successfully closed!)

THINKTANK Consulting

T: 778 995 4396

E : info@thinktankconsulting.ca

W: www.thinktankconsulting.ca