The leading cause of software project delays and missed schedules, according to Gartner, is poor software requirements.
Writing Effective Business Requirements

Mapping Vision to Realization

Presented by: Mr. Shimon Rothschild
Shimon@Shimon.us
Syllabus Objective

Corporate goals

The BRS

Prioritized business requirements

Effective business requirement

Idea
Cell Phone and other Distractions

We try to break for a few minutes every hour
Purely Practical

Russian Nesting Eggs

Requirements are like Russian nesting eggs, requirement within requirement.
Business Requirements

At the conclusion of these days you will be able to…

- Write an effective business requirement
- Capture and prioritize business requirements
- Maximize value of your efforts to validate requirement
- Identify business requirements to fulfill the bigger picture
Upon completion of this seminar, successful participants will . . .

- Write “good” business requirements
- Distinguish between business requirements and system requirements (specifications)
- Categorize requirements based on focus
- Identify and document business rules as requirements
- Group written requirements based on shared characteristics
- Evaluate a requirement for testability
- Discuss the difficulties in writing Quality, "-ability" requirements (e.g., reliability)
- Use templates to guide writing requirements
- Evaluate the completeness of your written business requirements

Focus on what needs to be done – not the do’s and don’ts
Read This!

What this courses defines as prerequisites for good business requirements

Upon completion of this seminar, successful participants will . . .

• Write "good" business requirements
• Distinguish between business requirements and system requirements (specifications)
• Categorize requirements based on focus priority
• Identify and document business rules as requirements
• Group written requirements based on shared characteristics
• Evaluate a requirement for testability
• Discuss the difficulties in writing Quality, "ability" requirements (e.g., reliability)
• Use templates to guide writing requirements
• Others can evaluate the completeness of your written business requirements
Three Days of Immersion

- Day 1: Purely Practical Requirements
  - Requirements in context
  - Define the business requirement specifications (BRS)
  - Characteristics of an effective requirement

- Day 2: How to Document

- Day 3: Verify and Beyond
Three Days of Immersion

- Day 1: Purely Practical Requirements
- Day 2: How to Document
  - Capture the requirements
  - Working with the BRS Template
- Day 3: Verify and Beyond
Three Days of Immersion

• Day 1: Purely Practical Requirements
• Day 2: How to Document

• Day 3: Verify and Beyond
  • Expectations and Measurements
  • Validate the FRS
  • Beyond Business Requirements
To Think or Just Write the BRS?

A stable / simple system is 25% pre BRS writing, it goes up from here.
Tales From the Trenches

Where is the documentation? Oh, so that’s why it’s failing
Purpose: Overview of gathering needs to write a good requirement

- Requirements in context of the development process
- Define the Business Requirement Specification (BRS)
- Characteristics of Effective Requirements
The Cost to Fix Bugs

The cost of fixing poor business requirements is SIGNIFICANTLY more expensive (if possible) than fixing programming errors.
Define Business Requirements

Changes in work activities / practices to:

- Enhance existing business activities
- Meet changing business objectives

They will often require new or radically modified workflows.
Are Requirements Important?

Requirements are the result of analysis

- Creativity - Think in new ways, not just improve
- Strategy - Fundamental part in formulation
- Essential prerequisite for:
  - Requirements specifications
  - System development
Key Questions

- Will you focus on business requirements, or user requirements?
- Will you focus on “low picking fruit” or highest return on investment?
- How will you test assumptions and validate results?

These decisions will be pillars for the duration of the project.
Why Projects Fail
The #1 cause is…

- Define Failure
  - Late
  - Expensive
  - Incomplete

Poor communication is the #1 cause, and communication between product and development is the most expensive.
Just a crack...

Not a break
Tales From the Trenches

Artificial intelligence isn’t…
Generic Development Process

Every ‘project’ goes through the following phases (states).
1. Define the business need
2. Describe the business requirement
3. Implement the solution
4. Verify the implementation meets the business requirement
5. Deliver to the to customer

Simplified “State Transition” Diagram
Iterative Development Process

Key differences:
• The work ‘chunk’ is smaller
• The cycle is 1-3 months
• The process repeats
• Requirements are ‘stacked’ and business need is always updated
Our Development Cycle

Account request

[done] [done] [done] [done]

Project charter → Requirement list → Prioritization → Groups → BRS Document

[ready]

Software design → Software dev → Internal UAT → External UAT → Delivery

[done] [done] [done]

Is it waterfall or iterative?
A Development Cycle

Account request
- Project charter [done]
- Requirement list [done]
- Prioritization [done]
- Groups
- BRS Document [ready]

Business need
- External UAT [done]
- Internal UAT [done]
- Software dev [done]
- Software design [done]
- FRS Document

Delivery
- [done]

Prioritization Groups
- done
- done
- done
- done

Account request
- Business need

Requirement list

Project charter

Prioritization

Groups

BRS Document

External UAT

Internal UAT

Software dev

Software design

FRS Document

2 times per year
4 times per year
8 times per year
The stakeholder was absolutely certain…
80 / 20 Rule

- Not everything is equal sort by bucket
- Let some one else prioritize within the bucket
- Define the minimum acceptable deliverable

80% effort on the most important requirement and 20% on the remaining
What are Buckets?

Collection of equivalent value items
Any item in the bucket can be selected at random
Reduces up-front planning time

Buckets do not capture dependency, therefore reduce dependencies
Tricks of the Trade

1. Prioritize requirements
2. Define buckets
3. Requirements to bucket
4. Buckets do not have to be equal

How many buckets are enough?
Exercise: Create your Own

- Each person think of a project they worked on or are working on.
- List 4-6 improvements that you want in the project.
- Project name and list on white board
- Together group requirements:
  - Critical
  - Medium
  - Low
Imagine getting a knock on your front door in the middle of the night. "**You have 10 minutes to leave your home**," says the sheriff, and you don't doubt him.

What possessions would you take with you?

We will continue this when you return from the break
What do Others Save?

- Australians
  - 50% family heirlooms
  - 25% laptop computer
- UK
  - 60% wallet - North
  - 80% photos - East
  - 30% cell phone - London
- USA
  - 25% would not leave
What is the BRS?
Business Requirement Specifications

What the business wants
Why the business wants it
How success will be measured

This session covers:
• Steps to identify business objectives
• Refining business objectives
• Write objectives to be reviewed
Tales From the Trenches

Business Objective: The game in every classroom.
Defining the Business Objective

- Business requirements are to advance a business goal
  - Tactical plan towards a strategic goal
  - The objective must align with a goal
- Business objectives might be vague or conflicting
  - Test that your business requirements are consistent with the goal and other requirements.
• The well being, **productivity, and cost issues that our solutions** addressed remain absolutely relevant regardless of the economic environment.

• Refocus the industry away from providing episodic fee per service encounters with individuals to an **outcomes approach built on solutions to improve well-being**

• a real change on the part of employers’ willingness to consider benefit designs and **engage employees more directly in managing their health** and a real appetite for care management from employers.

Are your requirements supporting, peripheral, or opposing these goals?
Sources for the Requirements

- Managers
- Customers
- Competition
- Your own vision

The requirement should always map back to a business objective
Reminder... What not How

Not How…
It is NOT about documenting how to implement …

It’s What…
It’s about ensuring that any implementation meets business requirements
Sample What and How

- The web site must allow users to find all CDs under $10 (WHAT)
- The web site must use faceted classification to allow browsing to CDs within specific price ranges. The links should be displayed in 12 pixel Verdana and have no underlines. (HOW)

As a __________, I want to __________ so that I can __________.” is a great way of understanding whether something really is a “what” or a “how”.
Capturing Customer Needs

1. **Customer:** This is what happened

2. **You:** And then what happened?

3. **You:** What else might happen?

4. **You:** What happened before it happened?

5. **You:** What else might happen before it?

The customer defines a point in space, capture the before and after
Purely Practical
Enhanced Reporting

1. **Customer:** I want enhanced reporting.

2. **You:** How will the data impact decisions?

3. **You:** What else might you use reporting for?

4. **You:** What caused a need for the data?

5. **You:** What else might need reporting?

The customer defines a need, you capture an opportunity and pain
Just a crack...

Not a break
Exercise

Write answers you might expect to hear from the customer.

**Customer:** I want better search results. I searched heart pain and the results were heart attack. It should have been about heartburn.

1. And then what happened?
2. What else might happen?
3. What happened before it happened?
4. What else might happen before it?
Logical Decomposition

- Incoming requirements are at different levels of granularity.
- If a ‘requirement’ can be broken into multiple, independent requirements do so.
- Try to eliminate dependencies between requirements.

After decomposition, then begin prioritization.
Exercise Decomposition

Problem: Make a better search engine for common causes of symptoms.

The following slide has some ‘business requirements’.
Decompose the Following

**Business Need:**

- Significant effort can result in significantly greater payback for a custom algorithm optimized for health care. The algorithm shall work with internal (licensed) content and the broader web experience.
- Integration of member provided data (HRA) shall provide a better result set than any general search algorithm (customized user experience).
- HRA or other questionnaire shall solicit answers to questions that can be used to provide a better search experience.
- Search shall be integrated with phone call coaching to notify user that a phone coach is available to discuss the problem.
- All query strings shall be saved with a map of how the user navigated. We shall use this to discover trends and prediction based on analysis.
- We shall standardize on the same standards (WHO International Classification of Diseases and others as defined).
- Integrate member tools to help the user better identify the cause (IE heart attack — how do you feel after walking 100 steps).
- Because content might be very important, consider buying vs. licensing the content.

**Key points:**

1. 80% of US population has searched web for health information.
2. Web search algorithms are optimized so that the most likely viewed documents are most often ‘worst cause’ and not typical cause.
How Long will this Take?

- Start – Ideas for requirements
- End – Prioritized, decomposed list of requirements
How Will the Requirement Look?

- Not bullet points
- Each requirement 3-5 sentences
  - Intro
  - Body
  - Conclusion
- Think ‘sound bite’ with substance

Group them by importance and document any dependencies
For Whom is Your Document?

Intended audience of the BRS:

- Peer review
- The BSA
- Corporate stakeholder
- Client stakeholder
- Secondary
  - Q/A
  - Development

It is NOT merely a prioritized decomposed list of requirements
Validation Check

Critical to validate – the cost of an error is too high to ignore.

- Peer review – not spelling and grammar, is it consistent with other projects?
- Stakeholders
  - TO DO list
  - Verify assumptions and
  - Challenge inconsistent requirements.
- BSA – Is it complete enough for YOU?
Break
For Lunch
Characteristics of the BRS

What makes effective requirements?

Guidelines
Writing style
Communicate

This session covers:
• Checklist of what to cover
• Best practices in writing
• Improve communication
IEEE Guidelines

1. Correct
2. Unambiguous
3. Complete
4. Consistent
5. Ranked
6. Verifiable
7. Modifiable
8. Traceable
Tales From the Trenches

Software “drive-by” shootings: A bad idea
Correct

Each requirement described as desired.

- Is this what you want?
- is this what you mean?

There is no tool or procedure to ensure correctness
Unambiguous

• English is inherently ambiguous – define ambiguous terms and be consistent

• Use domain known language – define domain language terms

• Business requirements define and use
  • Object: person, place or thing
  • Process: Transformation to another state
  • Behavior: Stimulus – Response (event / alarm)

In diagrams and documentation, use the same word for the same meaning
Complete

- All significant requirements for the current implementation
- Definition of all responses to realizable types of input data
- Definition of terms dictionary
- Polished document
- TBD includes who and what to eliminate

Complete according to the state in the work flow.
Consistent

- Handle real world inconsistencies
- Resolve document inconsistencies
- Inconsistencies in process
- Use of consistent terminology
Ranked

- Importance: High, medium, low
- Risk: Impact of failure

Prioritize them by company importance, then group them. Have engineering prioritize them within a group by cost/ risk
Verifiable

A requirement must be testable and the test must be understood

What should be done with a requirement that can’t be tested?
Modifiable

- Requirements change during the process
  - References are updated
  - Redundancy is avoided

This is especially problematic as the document is revised
Traceable

- Backwards traceability
  - Company objectives
  - Project charter
  - List of requirements
- Forward traceability
  - BRS
  - Q/A test plan
  - UAT

Assign ID as late as possible and use a schema that is not merely sequential
Just a crack...
Not a break
Exercise: Track Your Project

<table>
<thead>
<tr>
<th>Guide</th>
<th>Importance</th>
<th>Completed</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unambiguous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ranked</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verifiable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifiable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traceable</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Work in pairs or threes to discuss each project
Rate importance 0 (Low) to 10 (high)
Estimated how much has been completed of that guide (0-10)
Score is Importance * Completed
Best Practices in Writing

- Descriptive titles (noun-verb-noun)
- Annotate with notes
- Emphasize the **key points**
- Bullet points are OK
- Highlight assumptions
- Highlight “TO-DO' items”

Ask, what will be picked up first by scanning, is it most important?
Communications

- Utilize milestone deliverables to validate
  - Direction
  - Assumptions
  - Unknowns (TO DO items)
- Spreadsheets not Email to track issues
- Track document revisions

Scuba diving is in pairs for safety. The rule is valid in writing business requirements too.
Summary of Day 1

- 25 – 75% of the work is done before writing the BRS
  - Gather and refine the requirements
  - Verify the requirement is correct for the organization
  - Prioritize and group the requirements
  - Work intensely on highest priority requirements

- Best practices in writing the BRS
  - Utilize the IEEE standard for a sanity check
  - Use best practices in writing
  - Communicate and brainstorm before writing the BRS
  - Work with a partner

Writing a good requirement is the result of good research and communication
The leading cause of software project delays and missed schedules, according to Gartner, is poor software requirements.
Purely Practical

Objects, relationships and responsibilities

If requirements arrive as behavior, why don’t they develop software that way?
Day 2

Purpose: How to map business need (pain / opportunity) to engineering

- Capture and validate that the need is genuine
- Prioritize the need
- Document the need (using the template)
Capturing the Requirements

- **Stage 1:** Proposed system
  - Identify real world objects
  - Draw domain model

**Milestone Check**

- **Stage 2:** Define the proposed solution
  - Use cases and use case groups
  - Use case business requirements
  - Use case node business requirements

**MILESTONE**

If you need, get help from an engineer, and never skip stage 1
Stage 1: Real World Objects

- **Person, place or thing**
  - They have properties that change
  - They are catalysts for change
- **Often map to data tables**
  - Things someone wants reports on
  - CRUD – Create, read, update and delete

First step is to identify the objects that will be manipulated or act
Objects: Content Encyclopedia

Person, place or thing

- Healthwise ‘document’
- Language
- Category
- Catalog
- What else?

Abstract object: Many objects have similar properties, capture the similarity
Exercise: Define Some Objects

In your project, find objects that:

- CRUD (create, read, update or delete data about it)
- Interact with the system
- Are changed by the system

Take every one’s objects and build a library of objects
## Sample Objects

<table>
<thead>
<tr>
<th>People:</th>
<th>Member</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coach</td>
</tr>
<tr>
<td></td>
<td>Editor (content)</td>
</tr>
<tr>
<td></td>
<td>Lab</td>
</tr>
<tr>
<td>Documents:</td>
<td>Health record</td>
</tr>
<tr>
<td></td>
<td>Fulfillment content (encyclopedia article)</td>
</tr>
<tr>
<td></td>
<td>Lab result</td>
</tr>
<tr>
<td></td>
<td>Contact record</td>
</tr>
<tr>
<td>Other:</td>
<td>Health tracker</td>
</tr>
<tr>
<td></td>
<td>Risk calculator</td>
</tr>
</tbody>
</table>

Check list and ask does this solution interact with these objects?
Building a Domain Model

- Define your objects
- Define associations between objects
  - Not always is there a specific association
  - No orphan objects
What is a Domain Modeling?

Definition:

- A representation of how ANY business would handle the generic activities in the business
- An INCOMPLETE business object model focusing on explaining products, deliverables and important events

OO Programming is different, it focuses on the object entities first
Validation is Necessary

- Detect incomplete / incorrect information
  - Validate assumptions
  - Add missing associations
  - Identify irrelevancies
- Discovery
  - New methods
  - Plan for future

BSA should build the model, but only the product manager can validate it.
Coach Contact with Member

- **Objects**
  - Coach
  - Member
  - Fulfillment document
  - Contact record

- **Associations**
  - Coach talks to member
  - Coach sends a document
  - Coach wants a report of all conversations with member
Circles are objects, boxes are associations, but there are other ways too.
What report would be more difficult for engineering to create?
If you may need a report, define the association
Domain Model for Encyclopedia

Abstracting will significantly speed development and reduce cost.
Examples of Objects in the Domain

- Business Document: “How to quit smoking in 14 days”
- Language: English, Spanish, Simplified Mandarin
- Category: Smoking, Lung Cancer, Addiction
- Document Catalog: Longer Living, Frugal living
- Review: There are no “reviews”
  - Reader review: Billie’s review, Bobbie’s review, Joe’s review
  - Editor review: Editor A, Publisher P
Stage 1 Milestone

**Objects** - They are the topics of action in storyboards
- Actors
- Data
- **Domain model** – Simplifies the number of storyboards
  - Abstract instead of multiple storyboards
  - Different sub objects might be storyboard extension

Use the objects to create the definitions in the storyboards
Tales From the Trenches

Lip synchronization in animation
Stage 2: Use Cases (Storyboards)

Define the proposed solution

- Use cases and use case groups
- Business requirements from use cases
- Business requirements from use case nodes

What is the purpose of the use case?
Exercise: Answer the Following...

Define the proposed solution

1. What is the purpose of a Business Use Cases?
2. How is the Business Use Case defined, documented and described?
3. How and where do Business requirements fit in the process?
4. What and when is the best time to do Business Use Case Diagramming?
5. What are the Pro’s and Con’s of a Business Use Cases?

Business use case = storyboard
Just a crack...

Not a break
Suggested Answers

Define the proposed solution

1. What is the purpose of a Business Use Cases?
   Define who puts demands on it and who is interested in its output

2. How is the Business Use Case defined, documented and described?
   Defined based on business objectives
   Documents as a script and a diagram
   Described as sequence of actions that provides observable value to a business actor

3. How and where do Business requirements fit in the process?
   Required before the functional requirements are specified

4. What and when is the best time to do Business Use Case Diagramming?
   As soon as the key business requirements have been identified

What are the Pro’s and Con’s of a Business Use Cases?
   Not everyone can write or understand a storyboard, that’s why training

Business use case = storyboard
Ideas to Discover Storyboards

Storyboard sources:

- Each business requirement is a post condition to a storyboard
- Every object should be in at least one storyboard
- CRUD operations make storyboards
  - Create
  - Read
  - Update
  - Delete

The objective is a list of descriptive storyboard ‘titles’
Why Storyboards

In addition to standard requirements:

- Requirements
  - mix of high and detail requirements
  - Limited to text
  - Verbose

- Storyboards
  - Chain requirements to a bigger picture
  - Include corner cases
  - Diagram possible

Storyboards alone could provide sufficient documentation
Zachman Framework

Storyboard = business use case
Storyboard ≠ System use case
Key Differences

Business Requirements
- Detailing Business Workers and Entities
- Finding Business Actors and Use Cases
- Capturing a Common Vocabulary

System Requirements
- Expanding on the business requirements
- Abstracting commonality
- Identification of Dependencies

Business requirements - Row 2. System requirements – Row 3
Is a storyboard a use case?

Storyboard

- *Analysis* - The detailed definition of requirements for a particular area of the business.
- *Model of the business (Owner's view):* This defines -- in business terms -- the nature of the business, including its structure, functions, organization, and so forth.

Use case

- *Design* - The specific application of technology to the requirements defined during analysis.
- *Model of the information system (Engineering view):* This defines the model of the business, but in more rigorous information terms.

HINT: Write a storyboard about how users will use the system and what the system will do in response.
Storyboard organization

Before detailing the storyboards:

- Group storyboards by ‘logical’ business process.
- Prioritize groups and prioritize storyboard within group
- Identify objects in each storyboard
- Write for each storyboard:
  - Post condition
  - Five sentence description
  - Pre condition

Now storyboarding can begin
Five Sentence Description

- Sentence 1: Introduction
- Sentences 2-4: Body
- Sentence 5: Conclusion

If it takes more than 5 sentences, maybe it’s multiple storyboards
Storyboard Design

Before detailing a storyboard:

- The post condition, the requirement
- Five sentence description of the storyboard
- Pre condition

Steps:

1. What happens ("sunny day")
2. What could have happened ("rainy day")
3. What happens next ("link to another storyboard")
4. Why caused it to happen ("pre condition")

Why is pre condition twice?
Storyboard Parts

Steps:

1. General description (5 sentence paragraph)
2. Pre condition
3. Post condition
4. Actors
5. Assumptions (optional)
6. Storyboard story (“sunny day”)
7. Extensions (optional- alternatives or exceptions)

1-4 have already been done, merely include them in the template
Active Voice Writing
Subject – verb – direct object

Active
Harry
[subject]
throws
[verb]
the baseball.
[direct object]

Passive
Baseball
[object]
thrown
[verb]
by Harry
[subject]

Active voice helps identify actors in the use case
Write a Storyboard
Write in the active voice

Parts:

1. Title - descriptive
2. Each line - what happens, then what
3. Intersections – mark it then continue main flow
4. End when you have reached a business requirement
5. Optional – return to intersection to write flow

Active voice: Subject – verb – direct object
Storyboard “Gotcha’s”

Careful to avoid these errors:

1. A step is too detail
   1. Field delineation
   2. Message text
   3. Button location

2. A step not detail enough
   1. Actors not sufficiently identified
   2. Overloaded descriptions (“a page”)
   3. An implied assumption, not generally recognized,

Cliché: If it was easy, everyone could write a storyboard, so practice
Early Prioritization Requirements

1. Prioritize use cases if all costs were equal
2. Identify actors and objects that are low priority
3. Ask BSA for cost range to implement use case
4. Reprioritize based on cost range
5. Pick the first use case as a source for the BRS
   1. Most important
   2. Least expensive (but necessary)
   3. Most / least interdependent on other use cases
   4. Recommendation from BSA
6. “Run” it through the BRS Template

Get a sanity check from a co-worker or the BSA early in the process
Using a BRS Template

**Key components**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Environment</td>
<td>8. Rules</td>
</tr>
<tr>
<td>5. Features</td>
<td>10. Constraints</td>
</tr>
</tbody>
</table>

Most has been completed prior to starting the BRS Template
Glossary

**What is it:**
Minimum- the dictionary of words that can be misunderstood.

Best – list of words from *Corporate* dictionary that are referenced in the BRS

**How to complete it:**
- Most of the object model names should be included
- Words that are specific to the problem domain and not generally used outside.
- Words that have different meaning in other problem domains
- Words that are synonyms

If there is no corporate dictionary, create your own
Actors

What is it:
‘Role’ and ‘job description’. These are the people and external systems that appear in the storyboards.

How to complete it:
- Every actor is in the object model, but not every object model is an actor
- Actors make decisions
  - Initiator of storyboard always an actor
  - Other process in storyboard that could change the storyboard path

Do standardize, same role name in storyboards and don’t use same name for different roles.

System under discussion (SUD) in never an actor
Environment

**What is it:**
Describe the different ‘stages’ for the actor.

As data is more wireless, this becomes more important.

**How to complete it:**
Build a checklist of possible environments and then test if the environment is applicable

- Work
- Home office (off site)
- Mobile (cell phone, PDA)
- Custom station (special software required)
- None or limited connectivity
- Minimum expected hardware, software, etc

The environment is defined for EACH actor
Requirements (Business)

**What is it:**
The system shall…

It is a single sentence that describes what is desired. When presented with a candidate solution, it **MUST** be possible to test that it meets the condition.

**How to complete it:**
In most conditions a single 5 sentence paragraph is sufficient.

- Introduction sentence – What initiates the requirement
- 3 sentences – what happens
- Last sentence – the expected outcome

Every requirement must be verifiable.
Feature Set

What is it:
Pre condition and key events in the storyboard

A feature is a ‘junction’ in the storyboard. The precondition is always a feature because it is possible not to enter the storyboard.

How to complete it:
• Pre condition – what are “gatekeepers” to allow starting the storyboard
• Junctions in storyboard where alternative paths are open. Test each node if conceivable alternative path.

Sometime features (junctions) are identified after the storyboard is done
Storyboard

What is it:
Step-by-step a description of the vision of how the actor gets from pre condition to post condition

How to complete it:

- Actor does this…
- System does this (or different actor)

This keeps going until Actor reaches the objective of the storyboard

Identify, and optionally describe, points where the system or an actor might do other than as portrayed in the storyboard

Better not enough detail and add detail then write too much too soon
Non-functional

What is it:
Qualities of the system that generally cannot be measured.

Main categories:
• Execution
• Evolution

How to complete it:
• List ALL requirements that are desired, but cannot be measured.
• It is a TO DO list:
  • Desired and cannot be measured
  • Can someone else provide measurements?
• Common with external influence and no control
  • Government regulations
  • Technology

What is ‘non-functional’ in the BRS, could become functional at some time
Rules (Business)

**What is it:**
High level directions of what the organization should do

General principles that are used in the BRS

**How to complete it:**
- What existing business rules are applied in the BRS
- What new business rule has been declared
- What implicit business rule is implemented

Business rules transcend technology and the BRS, they are global
Processes (Business)

**What is it:**
A group of common storyboards to describe a process

**How to complete it:**
- An objective should be that a BRS is for a single process, not multiple processes
- For each pre condition and post condition, test if there is a logical storyboard that would connect into / or out
- Some input / output might be borders with other processes, identify these

This is mostly subjective, work with the engineer to decide the scope
Constraints (Business)

**What is it:**
A showstopper. Any constraint should be demonstrated to be mitigated or addressed VERY EARLY in the process.

**How to complete it:**
- Identify show stoppers and if there is an acceptable work around
- Verify the constraint with stake holder, maybe it is merely a very nice to have
- Define when and what condition must be met to proceed

The number of constraints indicates bleeding, leading, or trailing edge.
Writing a Storyboard

Sample Storyboard:
1. Make breakfast
2. Sit down
3. Eat breakfast
4. Clean up

Sample requirements use case: Clean Up
1. Bring dishes to sink
2. Wash dishes
3. Dry dishes
4. Put dishes away

Sample design use case: Wash Dishes
1. Turn water on to 25 PSI pressure with equal measurement hot and cold water
2. Apply ¼ oz. liquid dish soap to blue sponge
3. Apply soapy side of sponge to dish and move sponge clockwise beginning in the center, until outer edge covered with soap.
4. Hold dish under faucet for 10 seconds at a 45-degree angle to the sink then turn water off.
Exercise

Write a storyboard going to a restaurant
Write a functional use case
Scoping the Storyboard

Diagram:

- **Work**
  - To Highway
    - Leave Parking
      - Right Hardy
        - Left Elliot
        - I10 Westbound
      - Right Ocean
    - Drive
      - From Highway
        - Park on left
          - Right Broadway
Inserting Business Rules

Storyboard: Pick up pet from the kennel

Business Rules:
1. Owner must pay before delivery
2. Verify payment before delivery

Would this business accept PayPal?
Prioritize the Requirements

Simple formula can be applied to determine priority:

\[ I \times C \times R = \text{priority} \]

I: Importance - Not all Requirements are equal
- High, medium, low (business, no cost / risk associations)

BSA to provide
- C: Cost (High, medium low)
- R: Potential cost overrun (risk: High, medium, low)

Decimal value 0 to 1 (low to high)
Prioritize Sample

Requirement A:
I = High C= High and Risk = High
\[0.8 \times 0.2 \times 0.2 = 0.032\]

Requirement B:
I = High C= Medium and Risk = Low
\[0.8 \times 0.5 \times 0.8 = 0.32\]

Requirement C:
I = Medium C= Low and Risk = None
\[0.8 \times 0.8 \times 1 = 0.64\]

Should requirement “C” be completed before “A” and “B”?
Purely Practical
Walkthrough Building the Lab Feed BRS

Steps
1. Set meeting for interview
2. Interview
3. Validate overview
4. Write first pass
5. Review first pass
6. Write second pass
7. Submit for approval
Purely Practical
Step 1: Set meeting

- Ready conference toolbar on Outlook to easily schedule and send invitations
- Provided list of questions
- Discussion of storyboards
- Participation optional to engineering
Purely Practical
Step 2: Interview

- Participants (BSA, Product manager, Engineer)
- Discuss question
  - Delivery date (no milestones asked!)
  - System data input
  - System data output
  - What is the objective (map to corporate goals)
- Product manager encouraged to walk through storyboards
Purely Practical

Step 3: Validate Overview

Based on objective, a five sentence overview of project.

This is done for early validation that the work will be aligned to goals.
Purely Practical

Step 4: First Pass

- Approved overview
- The objects (system, member, lab, coach)
- The domain model using 3 of 4 objects (unsure about coach)
- 4 storyboards based on stories provided by product manager
  - 2 member experience
  - 1 system point of view
  - 1 lab point of view
- 8 requirements derived from the storyboards

The document was submitted for review
Summary of Day 2

- Only as good as it is accurate
  - Define the objects
  - Define the associations
  - Create storyboards
  - Prioritize your efforts before
  - Get BSA cost input to reprioritize based on budget / time constraint
- Write the BRS
  - Object model and use cases are sources
  - Use the template

Heuristic: 50% of your effort is before writing the ‘BRS’
The leading cause of software project delays and missed schedules, according to Gartner, is poor software requirements.
Day 3

Purpose: More mapping business need and solution validation

- Expectations and Measurements
- Validate the FRS
- Beyond Business Requirements
Expectations and Measurements

- Acceptance Test is part of the requirement
- Threshold of accuracy is required
  - How well must corner cases be covered?
  - Can accuracy be incremental?
  - Is there a cost / accuracy range?
- What is the impact of failure?
Expectations

- There may be many expectations from a single business requirement
- Define under what conditions the business requirement is expected to be called
- What inputs
- What outputs
Noun – Verb – Noun is Testable

- Noun – Verb – Noun usually defines:
  - Noun (1) – who
  - Verb – does
  - Noun (2) on/to what

- Map N-V-N to Testing
  - Noun (1) provides input
  - Verb is the business requirement
  - Noun (2) provides output
Is this Testable?

*Business Requirement: PopWorks (Embrace) shall have the ability to identify IWP members.*

Steps:
1. Write some N-V-N statements to describe the requirement
2. Define the range of input and out values for each N-V-N

Example:
NVN: Embrace-identifies- IWP member

Tests
- Embrace has name and ‘somehow’ identifies name is IWP member
- Embrace has name and cannot identify name is IWP member
- Embrace has name and ‘thinks’ name might be IWP member
Exercise: Is this Testable?

Business Requirement: PopWorks fulfillment items to be retrieved from the IWP member’s message center.

Steps:
1. Write some N-V-N statements to describe the requirement
2. Define the range of input and output values for each N-V-N

You try, hint the requirement is also ambiguous, identify how.
Sample Solutions

Business Requirement: PopWorks fulfillment items to be retrieved from the IWP member’s message center.

Steps:
1. Write some N-V-N statements to describe the requirement
2. Define the range of input and out values for each N-V-N

Example:
NVN: Embrace-sends-item

Tests
- Embrace sends item and can be mapped to IWP member
- Embrace sends item that cannot be mapped to IWP member
- Embrace sends item and IWP does not know how to link to fulfillment item
Just a crack...

Not a break
Tales From the Trenches

Only 3500 usability requirements?
Threshold for accuracy

- Only human safety may require 100% accuracy, otherwise it is too expensive
- Often the increase from 80 to 90% is equal to the cost from nothing to 80%
- Sometimes human intervention is better than attempting to obtain near 100% accuracy
- Accuracy can be defined by cost (what does $10,000 get me?)

Not every requirement should define the threshold, but the most expensive and most important should
How Accurate do you Need to Be?

If you don’t define the accuracy requirements, someone else will. It’s your budget, do you want it spent on misplaced accuracy?
Tales From the Trenches

Orders from Buy.Com
How Accurate do you want it?

Business Requirement: PopWorks (Embrace) shall have the ability to identify IWP members.

100% accuracy, every PopWorks order shall be accurately identified if it is associated with a IWP member.
80% accuracy, some IWP members not identified no wrongly identified as IWP
60% accuracy, some wrongly identified as IWP

The % value is less important than the idea that inaccuracy is acceptable.
Exercise: How Accurate will it Be?

Business Requirement: PopWorks fulfillment items to be retrieved from the IWP member’s message center.

Steps:
1. Write down some of the conceivable losses, what is acceptable loss?
2. Estimate how much (what %) of your total budget you want to spend on this feature.
3. Define the worst case loss, beyond that you would not have it implemented.
4. Define max % you would spend on the feature (at worst case loss)

100% of all IWP members getting 100% of all messages is expensive, do you really need that? Would some ‘lost messages’ be acceptable?
Validate the FRS

- Is it what you meant?
  - Does the specification meet your implicit vision?
  - Is the implicit vision (your requirement) what you really want?
- Is it prioritized correctly?
- Did the FRS answer open questions in the BRS?
- Are assumptions correct?
- What can be demonstrated on paper? (flows, screens)
  - Ideally there will be a virtual ‘walk through’
  - Is the right data being captured and retrieved?

It’s like the floor plan of the building, next is construction
What is in the FRS

Not every FRS has everything

- Cover only what is needed
- Not all components necessary in the FRS

Components of the FRS

- Text to elaborate on the BRS requirement (mapped)
- Functional use cases
  - Text
  - Diagram
- Diagrams
  - State transition
  - Activity
  - Object / relation map (ORM)
  - Data flow
  - Database object / relation
FRS Prioritization and Open Issues

- High, medium and low priorities
- Reevaluate based on cost and risk
- Open issues can impact prioritization

Consider carefully changes to the priorities, BRS is business and the FRS is more aligned with engineering.
Steps to Validate the FRS

1. Does the text expand on the vision or diverge?*
2. Answer open issues and confirm assumptions
3. Walk through use cases*
4. Walk through state diagrams*
5. Walk through activity diagrams*
6. Are the object / relation map associations correct?
7. Demonstrate how data is queried for reporting
8. Any timing and conditional delays correct (data in)

* Four ways to depict differently the same requirement
Step 1: Is it Your Vision
Expanding on the BRS

Sanity check:
- Does it describe what you envisioned?
- Are assumptions correct?
- Who must answer open issues?

Caution:
This should be done early, frequently and documented:
1. Start with use cases (avoid fancy formatting)
2. Track comments (spreadsheet works well)
Step 2: Capture Open Issues
Discover of open and conflicting ideas

Encourage assumptions:

- Document for review
- Explanation of why assumption
- Alternative to an assumption

Assumption making is significantly faster than question and answer. Basic Q/A should have happened in step 1
Step 3: Use Cases
Who interacts with what and how

Use Cases:
- Actor based work flow
- Only two outcomes, success or failure
- Ping pong, not volleyball or soccer

Generally unproductive to review text when diagram is available. The text is useful in step 1, the diagram is better in step 3.
Sample Use Case Diagram

Input order:
- Request content
  - IWP gets notification of content request
  - Filter for IWP member list
    - Non member list
    - IWP member list
      - Generate messages
        - Message list
          - Email
          - Email with hyperlink
          - Create Email
          - Messages
Step 4: State Diagrams
Moving from state to state

States and transitions:

- State is a snapshot, static
- Transition is the transformation from one state to a different
- States are analogous to maps and routes

If there is no line connect states, there is no direct path between them
Sample State Diagram

- Intro web page
- Cookie?
  - [no]
  - [yes] Cookie
- Country / language
  - [no language]
  - [no login]
- Login
  - [failure]
  - [language and login]
- [login]
- Logged in
Step 5: Activity Diagram

Time based flow of data

Activity Diagram:

- Data work flow
- Multiple outcomes (or no outcome)

Interface between actors is critical to be understood.
Sample Activity Diagram

Call center

Request

Harvest requests

Post request list

Digital request list

Driver

Non identified list

Formatted IWP member list

IWP member list

Filter for IWP members

Digital request list

Event: On data transfer

Call center messages

IWP

Email to member

Messages to member message center

Update database

Sort messages by member
Step 6: ORM Diagrams
Objects and relations map

ORM are closed systems:
• Every object relates to and is related to an object
• All objects are software objects
• All relations are describe how an object can identify the other

ORM describes how a system can generate reports
Sample ORM Diagram

- Member
  - ... selects .../... Assigned to ...
  - ... views .../... displayed to ...
- Language
  - ... Has a .../... writes ...
  - ... in .../... contains ...
- Content
- Publisher
  - ... Owned by .../... publishes ...
Remaining Steps: Data Validation
Data capture and reporting

Demonstration that data is available:

• Tables for data capture
• Foreign key constraints
• Data retrieval for reporting

If reporting is critical, then a walkthrough of data retrieval is necessary
Beyond Business Requirements

- Why projects fail
- Team effort
- Making milestones
- Fad, fashion and style
Why Projects Fail
2 of 3 projects software projects fail

Definition of failure:

1. Significantly over budget
2. Significantly late
3. Functionality does not meet minimum requirements

Numbers 1 and 2 are often because of changing requirements
Flexible Requirements
Clear and configurable requirements

Easy to interpret and easy to modify:
• There will be assumptions made – will they be correct?
• Misunderstandings during development will happen
• Business requirements do change during development

Provide a big picture for understanding why requirements might change
Team Roles

How each member contributes to BRS

Expectations:

- **Product owner**: The big picture
- **BSA**: The FRS
- **Engineering**: Prioritization from engineering POV
- **Q/A**: Use cases, especially extensions
- **Documentation**: Virtual ‘walk through’ of the use case

Better some input early than much input late
Milestones
How often and what to include

Milestone deliverables

- Milestone deliverables must be measurable
- Sometimes deliverables can be alternative order
- Define points where failure might end the project
- Allowable to change the requirement on milestone failure?

Make a milestone and then define expectations
Tricks of the Trade

1. Map requirements to process, not project
2. Keep running list of changes
3. Milestones for success are essential

The bigger the project, the more likely the failure
Mapping Requirements to Process

1. Process = use case (more or less)
2. Processes should be independent
3. Think parallel processing
The Spreadsheet

Each person independently submits questions / bugs for review.

<table>
<thead>
<tr>
<th>Document</th>
<th>Line ID</th>
<th>Submitted</th>
<th>Comment</th>
<th>Status</th>
<th>Response</th>
<th>Follow up</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-001</td>
<td>7</td>
<td>Billy Joe</td>
<td>can never be allowed</td>
<td>Accepted</td>
<td></td>
<td>to be fixed doc 1.1</td>
</tr>
<tr>
<td>UC-001</td>
<td>7</td>
<td>Bobbie Sue</td>
<td>will not happen</td>
<td>Rejected</td>
<td>duplicate</td>
<td></td>
</tr>
<tr>
<td>UC-001</td>
<td>10</td>
<td>Billy Joe</td>
<td>how can user exit from here?</td>
<td>Closed</td>
<td></td>
<td>fixed doc 0.4</td>
</tr>
</tbody>
</table>
Milestones for Success

1. Coordinate different inputs from different staff
2. Missing milestones should impact the BRS
3. Where business uncertainty, create milestone and junction of point of no return
Fad, Fashion or Style
Will the requirement go out of style?

The life of the requirement:

- Fad: Necessary, but soon replaced
- Fashion: The competition has it, or will have it soon.
- Style: Good taste, but taste evolves.

It might be critical requirement and still have a very short life.
The World Keeps Spinning

- Watch Paradigm Shifts
  - Facebook
  - MySpace
  - Mashup
- Cloud computing and SaaS
- Work environment and Agile

Changes are happening faster, lead or you will be pushed aside
Course Summary
Business requirements are most important

Not all requirements are equal:
• Importance
• Added value
• Timeliness

Not all documentation is equal:
• Document requirement importance
• Work mostly on most important
• Verify the team ‘gets it’
Summary

Do it right... and you won’t lose your shirt!

- BRS critical to success
- Not all requirements should be equal
- Design to verify and track progress