Requirements and user stories

15.1 Introduction

The importance of a well understood, prioritised and agreed set of requirements is self-evident. However, the attempt to define a full and detailed set of requirements too early in a project often proves to be counterproductive, restrictive and wasteful. It is not possible to define all of the detailed requirements at the outset of a long project. The business environment changes as time progresses; new requirements and opportunities present themselves. As the project progresses, the team understand more about the business need. Defining detailed requirements too early means either needing to change the specification later, which wastes the original work, or delivering to the originally-specified requirements and subsequently failing to adequately satisfy the business need.

DSDM acknowledges this dilemma and proposes a better way of working. DSDM advises the capture of requirements at a high level, early in the project. Further detail is gradually elicited as the project progresses, deliberately leaving the finer details as late as practicable, i.e. until the Evolutionary Development and the Timeboxes.

15.2 What is a Requirement?

At its simplest, a requirement is **a service, function or feature** that a user needs. Requirements can be functions, constraints, business rules or other elements that must be present to meet the need of the intended users.

For example:

In a training company with its own training centre:

- The Course Manager has a requirement to schedule training courses and reserve rooms, in order to make available courses visible and to ensure courses run effectively
- The Training Centre Manager has a requirement to keep track of what training is running, in order to ensure appropriate allocation of trainers to courses
- The Financial Accountant has a requirement to maximise the amount of time that the training rooms are in use, in order to maximise revenue from the rooms

If the product to be delivered is a custom-built car, the requirements defining this would be more feature-based:

√ A means of propulsion

√ A maintainable steering capability

√ A comfortable place to sit

However, it should be noted that the following are not requirements, but solutions:

X An engine

X A steering wheel

X Bucket seats

DSDM projects aim to state requirements in a manner which avoids tying them to a particular solution for as long as possible. This is because more flexibility can be retained in how a solution is eventually provided if requirements are expressed as what needs to be achieved, rather than how they will be met from a technical point of view, e.g. “a means of propulsion”, rather than “an engine”. A solution expressed too early may become a constraint on what can be achieved within time and budget.

15.2.1 Categories of requirement

The success of any solution is the product of two aspects:

- what it does (functionality, features)
- how well it performs against defined parameters (non-functional attributes, acceptance criteria, service levels)

15.2.1.1 Functional Requirements (FRs)

Functional requirements express function or feature and define ***what*** is required, e.g.

- Visit customer site
- Obtain conference venue

The requirements do not state ***how*** a solution will be physically achieved.

- Drive to customer site is one possible solution. However, fly to customer site or travel by train to customer site are potential alternative solutions which may be worth consideration
- Build conference centre is one possible solution. Hire a hotel room is an alternative solution Stating requirements early in the project as what rather than how allows room for flexibility and innovation later.

15.2.1.2 Non-functional Requirements (NFRs)

Non-functional Requirements define how well, or to what level a solution needs to behave. They describe solution attributes such as security, reliability, maintainability, availability (and many other “...ilities”), performance and response time, e.g.

- responding within 2 seconds
- being available 24 hours per day, every day

These NFRs may be:

- Solution-wide or impacting a group of functional requirements: e.g.
    - All customer facing functionality must carry the company logo
    - All customer-facing functionality must respond within 2 seconds to requests
- Related to a particular functional requirement, e.g.
    - Hire conference venue might have NFRs of accessibility, security, and availability

15.3 User Stories

15.3.1 What is a User Story?

A User Story is a requirement expressed from the perspective of an end-user goal. User Stories may also be referred to as Epics, Themes or features but all follow the same format.

A User Story is really just a **well-expressed requirement**. The User Story format has become the most popular way of expressing requirements in Agile for a number of reasons:

- It focuses on the viewpoint of a role who will use or be impacted by the solution
- It defines the requirement in language that has meaning for that role
- It helps to clarify the true reason for the requirement
- It helps to define high level requirements without necessarily going into low level detail too early

User goals are identified and the business value of each requirement is immediately considered within the user story.

User Stories are often deemed to comprise three elements - **the 3C's**

- **C**ard
- **C**onversation
- **C**onfirmation

15.3.2 User Story format

The format of the User Story is as follows:

As a < role>

I need

So that

These two examples demonstrate User Stories at different levels, but using the same format:

At a project level

As a Marketing Director,

I need to improve customer service

So that we retain our customers.

At a detailed level

As an Investor,

I need to see a summary of my investment accounts,

So that I can decide where to focus my attention.

User Stories provide another powerful message. Choosing User Stories to define requirements demonstrates an intention to work collaboratively with the users to discover what they really need. The User Story is brief and intended to be a placeholder for a more detailed discussion later – the Conversation. Much of the detail of User Stories emerges during Timeboxes as part of evolutionary development. High-level User Stories (Epics) are broken down by the Solution Development Team into more detailed User Stories just before development commences on that group of stories. Even then, the User Stories are not intended to be full specifications of the requirements. Fine detail may not need to be written down at all, but may simply be incorporated directly into the solution as part of the work within a Timebox.

The user story format helps to ensure that each requirement is captured in a feature-oriented, value oriented way, rather than a solution oriented way.

In DSDM projects, User Stories are recorded in the Prioritised Requirements List (PRL). This is the equivalent of a Product Backlog in other Agile approaches.

15.3.3 User Story – the Card

From the PRL, User Stories are often printed onto physical cards, for planning purposes and to help the Solution Development Team monitor progress.

The Front of the Card

On the front of the card, the following information is typically displayed:

- A unique “Story Identifier”, usually a number or reference
- A clear, explicit, short name or title
- “As a I need , so that ”; this section states:
  - who is the primary stakeholder (the role that derives business benefit from the story)
  - what effect the stakeholder wants the story to have
  - what business value the stakeholder will derive from this effect.

The Back of the Card

On the back, the Confirmation area contains:

- Acceptance criteria (the test criteria)

These acceptance criteria define, at a high level, the test criteria which will confirm that this user story is working as required. These are not intended to be the full test scripts, but will be used to expand into the appropriate test scenarios and test scripts during Timeboxes, as necessary.

For User Stories at the highest level (sometimes called a project Epic), the acceptance criteria may be used to define the aims of the project using criteria that may be measured after the project has completed (as part of the Benefits Assessment).

Project acceptance criteria example:

- Is customer retention improved by 20% within two years?
- Is product range increased by 10% within 5 years?
- Has speed of dispatch improved to within 24 hours of time of order for 99% of in-stock items within 6 months?

User Story Example:

Story Identifier: STK001

Story Name: Customer Order

Description: As a Customer, I need to place an order so that I can have food delivered to my house.

Confirmation: Acceptance Criteria examples:

Functional:

- Can I save my order and come back to it later?

- Can I change my order before I pay for it?

- Can I see a running total of the cost of what I have chosen so far?

Non-functional: availability:

- Can I place an order at any time (24 hours per day or 24/7/365)?

- Can I view the order at any time (24 hours per day or 24/7/365) up to and including delivery?

Non-functional: security:

- Are unauthorised persons and other customers prevented from viewing my order?

15.3.4 Well constructed User Stories

Bill Wake's INVEST model provides guidance on creating effective User Stories:

Independent	Stories should be as independent as possible from other stories, to allow them to be moved around with minimal impact and potentially to be implemented independently. If stories are tightly dependent, consider combining them into a single user story.
Negotiable	Stories are not a contract. They are "placeholders" for features which the team will discuss and clarify near to the time of development.
Valuable	Stories should represent features providing clear business value to the user/owner of the solution and should be written in appropriate language. They should be features, not tasks.
Estimable	Stories need to be clear enough to estimate (for the appropriate timeframe), without being too detailed.
Small	Stories should be small enough to be estimated. Larger "Epic" stories should be broken down into smaller User Stories as the project progresses. The stories after splitting still follow the INVEST criteria.
Testable	Stories need to be worded clearly and specifically enough to be testable.

A well-written user story is clear, concise and complete. Some simple checks are:

- It does not combine with, overlap nor conflict with other User Stories
- It conforms to organisational and project standards and policies where applicable
- It is traceable back to the business needs expressed in the business case and project objectives
- Where several User Stories relate to the same feature, but for different users, they are cross-referenced to each other

15.4 Requirements Through the DSDM Lifecycle

Projects need:

- A clear project objective
- A statement of the business vision
- A Business Case, agreed with key stakeholders

These form the anchor for the deliberate evolution of the more detailed requirements, iteratively and incrementally, as the project progresses. As the hierarchy of requirements emerges in expanding detail, as the project unfolds, each requirement/User Story can always be traced back to this original vision, as it evolves to meet the real and current business needs.

15.4.1 Requirements activity during Feasibility

All projects begin with an idea and an expectation of benefits t o give a return on investment. The Business Analyst ensures that the Terms of Reference (which is sometimes vague or unclear) is expanded to provide a clear project objective, a business vision and an outline Business Case . The project vision is clarified and key project objectives are defined. The highest level Epic User Story is the objective of the project. The User Story format can be effectively used to clarify:

- Who needs this? (Do we have the right Business Sponsor?)
- Why do they need it? (What is the key business value expected or needed?)
- What are their expectations? (What are the high-level acceptance criteria?)

The User Story format also helps to identify the key stakeholders with whom to gain agreement for the requirements.

In Feasibility, the User Stories (sometimes called Epics or Themes) should constitute a small number of clear statements that are just sufficient to scope the project, to identify whether it is worth proceeding further and to establish likely costs and benefits achievable. DSDM recommends typically less than 10 requirements/User Stories at this point.

Non-functional requirements (see above) may also emerge at this point, but these are expected to become clearer and more detailed throughout the project. Some of the more critical ones may be evident from the outset, when the project objective is established, and these need to be captured because they may constrain some of the choices for the project.

Even at this high level, User Stories help to focus on the value of what is required.

> For example:
>
> As a Human Resources Manager, I need a better way to deal with employee records, so that employee history can be tracked including their training and career moves.”

is a far more effective way of defining what the business needs, than the vague but technically constraining statement:

"The Organisation will implement a human resources system"

The user story format helps to bring out the real objectives of a major change.

15.4.2 Requirements activity during Foundations

During Foundations, more understanding of the requirements is needed, sufficient to clarify the scope of the project, prioritise, estimate and formulate a realistic Delivery Plan.

During Foundations, the high-level Epic or Theme stories established in Feasibility are now broken out into simple User Stories (functional and non-functional). User Stories defined by the end of Foundations in a DSDM project must be specific enough to estimate and small enough to fit into a Timebox (typically 2 – 4 weeks work). This is not the lowest level of breakdown that the project will achieve, but by the end of Foundations User Stories need to be just sufficient to allow for estimates of work to be done and to plan a schedule of Timeboxes for the first Project Increment.

At Foundations, User Stories are assembled into a Prioritised Requirements List (PRL). The focus is on describing the business need embodied in each User Story, in a way which does not constrain unnecessarily how the requirement will be achieved.

Key non-functional requirements should also be considered and documented during Foundations. It may be difficult or impossible to accommodate such requirements if they are discovered too late in the project.

The PRL is baselined at Foundations, to give a clear checkpoint for the set of requirements which was used for planning. In this way, new requirements which emerge during development are clearly identified, and their impact can be assessed.

15.4.3 Requirements activity during Evolutionary Development

At the outset of each Timebox, the User Stories allocated to that Timebox will be further investigated. The User Stories from the PRL are broken down into more detailed User Stories which are small and clear enough for the team to work from. The detail is only elaborated one Timebox at a time, and thus the complexity of the requirements is managed. Also, the fine detail is only elicited immediately before that element of the solution is created. This avoids time being wasted on developing detail on all areas up front.

During Timeboxes, the detailed requirements/User Stories emerge iteratively. The Business Analyst captures the appropriate level of emerging detail within the PRL, where this is not adequately captured within test criteria, prototypes and the Evolving Solution itself. The Business Analyst also works collaboratively with both Solution Development Team and project level roles to help retain the project’s focus on value and priorities.

New requirements may emerge which were not identified during Foundations. The Business Analyst facilitates the consideration and impact analysis of these and records their inclusion or otherwise in the project, based on discussions with the Business Ambassador, the rest of the Solution Development Team and/or Business Visionary. The Business Analyst also records details of, and reasons for, any lower priority requirements being de-scoped by team agreement during Evolutionary Development.

15.5 Conclusion

Requirements evolve and emerge in a DSDM project. Analysis of the detailed requirements is deliberately left as late as is sensible, to avoid unnecessary rework and to manage complexity. Because of this, it is important to obtain agreement to a high-level baselined set of prioritised requirements in the PRL in the early phases of a DSDM project. This gives scope, direction and an appropriate degree of control for the project to evolve the detail whilst allowing change to be embraced and controlled.