7

Managing Scope

Effective Requirements Management

- Team Skill 1 - Analyzing the problem
- Team Skill 2 - Understanding user needs
- Team Skill 3 - Defining the system
- Team Skill 4 - Managing scope
- Team Skill 5 - Refining the system definition
- Team Skill 6 - Building the right system

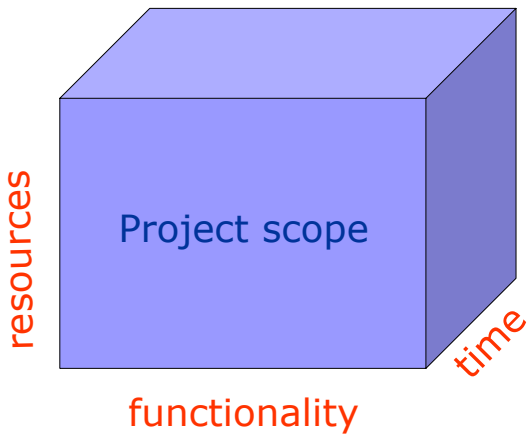
Managing Scope

- The Problem of Project scope
- Establishing Project Scope
- Managing your customer

The Problem of Project Scope

- What is project scope?
  - a combination of product functionality, project resources, and the time available
- In other words, project scope is a function of
  - The functionality that must be delivered to meet the user's needs
  - The resources available to the project
  - The time in which to achieve the implementation

Components of Project Scope



Functionalities of Project Scope

- The Functionalities of Project scope is classified into 2 parts:
  - Functional requirements – (data base, GUI, other business rules)
  - Non-functional requirements – (performance, scalability, reliability, security, etc.)
- The features of a project are referred to both Functional and Non-Functional requirements of the Project
- The PM has to measure at the start of the project “what is the amount of scope given to the development team by the management and the customer?”

A Few facts of Project Scope

- Brooks law:
  - If the effort required to implement the system features of the project is equal to the resources over the time available, the project has an achievable scope.
- Adding labor to a late software project makes it even later

Establishing Project Scope

- How to establish project scope?
- Listing a high level requirements baseline, an itemized set of features intended to be delivered
- Setting priorities of the features
- Assessing a rough level of effort required for each feature of the baseline
- Estimating the risk for each feature, or probability that implementing it will cause impact on the schedule and the budget
- Reducing the scope

The Baseline Requirements

- The baseline is the itemized set of features, or requirements, intended to be delivered in a specific version of the application
- For example, let's consider a shrink-wrapped software product with a list of following features:
  - Feature 1: External relational data base support
  - Feature 2: Multi-user security
  - Feature 3: Compatibility with Adobe
  - Feature 4: Implementation of tool tips
  - Feature 5: Compatibility with Microsoft, Linux, Unix
  - Feature 6: Compatibility with IE and Netscape

Prioritization

- Importance
  - 1/3/5
  - High/ Medium/ Low
  - Critical/important/Useful
- Risk
  - High/Medium/Low
- Effort
  - High
  - Medium
  - Low

Setting Priorities

- The priority scale is mutually decided by the development team and the customers
  - It may be sometimes decided by voting

Feature	Priority
Feature 2	Critical
Feature 5	Critical
Feature 6	Critical
Feature 1	Important
Feature 3	Useful
Feature 4	Useful

Assessing Effort

- The effort of each feature is decided by the development team
  - It may be sometimes decided by voting
  - It is based on the resources available (the experience of the development team) over the time available

Feature	Priority	Effort
Feature 2	Critical	Medium
Feature 5	Critical	High
Feature 6	Critical	Low
Feature 1	Important	High
Feature 3	Useful	Medium
Feature 4	Useful	Low

Adding the Risk Element

- It is the probability that the implementation of a feature will cause an adverse impact on the schedule and the budget
  - The development team establishes risk using the same low-medium-high scale
- There is often little correlation among priority, effort and risk

Feature	Priority	Effort	Risk
Feature 2	Critical	Medium	Low
Feature 5	Critical	High	High
Feature 6	Critical	Low	Low
Feature 1	Important	High	Medium
Feature 3	Useful	Medium	Low
Feature 4	Useful	Low	Low

Reducing Scope

- In projects, in order to provide a reasonable probability of success, it will be necessary to reduce the scope by as much as a factor of two

Feature	Priority	Effort	Risk
Feature 2	Critical	Medium	Low
Feature 5	Critical	High	High
Feature 6	Critical	Low	Low

Baseline (Features above this line are committed features)

Feature	Priority	Effort	Risk
Feature 1	Important	High	Medium
Feature 3	Useful	Medium	Low
Feature 4	Useful	Low	Low

A Thumb Rule for Baseline

- A thumb rule to draw the baseline at the critical requirements perhaps one or two important features

Feature	Priority	Effort	Risk
Feature 2	Critical	Medium	Low
Feature 5	Critical	High	High
Feature 6	Critical	Low	Low
Feature 1	Important	High	Medium

Baseline (Features above this line are committed features)

Feature	Priority	Effort	Risk
Feature 3	Useful	Medium	Low
Feature 4	Useful	Low	Low

Managing Your Customer

- It is how the development team actively engage the customers in managing their requirements and their project scope to ensure both the quality and the timeliness of the software outcomes
- Points to remember:
  - Customer who are a part of the process will own the result
  - Getting the job done right means providing enough functionality at the right time to meet the customer's real need

The Baselined Requirements