CS 4530: Fundamentals of Software Engineering
Module 2, Lesson 2
Requirements and Risk

Rob Simmons

Khoury College of Computer Sciences

Review: User Stories

- As a College Administrator, I want to keep track of students, the courses they have taken, and the grades they received in those courses, so that I can advise them on their studies.

As a <role>
I want <capability>
so that I can <get some benefit>

Satisfaction Conditions

- We will build a secure web application backed by a persistent database that allows an authenticated administrator to:
  - Add a new student to the database
  - Add a new student with the same name as an existing student.
  - Retrieve the transcript for a student
  - Delete a student from the database
  - Add a new grade for an existing student
  - Find out the grade that a student got in a course that they took

Writing User Stories: INVEST

- Independent
- Negotiable
- Valuable (has value to client)
- Estimable (able to estimate development effort)
- Small
- Testable

As a <role>
I want <capability>
so that I can <get some benefit>

Learning Goals for this Lesson

At the end of this lesson, you should be able to

- Explain the overall purposes of requirements analysis
- Recall the three major dimensions of risk in requirements analysis
- Explain the connection between requirements analysis and user stories
- Identify functional and non-functional requirements, and give examples of each

Where does requirements analysis fit in?

	PEOPLE	PROCESSES	PROGRAMS
PLANNING	Design problem: what user stories meet the needs of **waves hands at everyone, everywhere**		
ORGANIZING	Design problem: what conditions of satisfaction meet the needs of a given user story?		
IMPLEMENTING	Design problem: what programs meet the needs of a condition of satisfaction?		

Where does requirements analysis fit in?

	PEOPLE	PROCESSES	PROGRAMS
PLANNING			
ORGANIZING			
IMPLEMENTING			

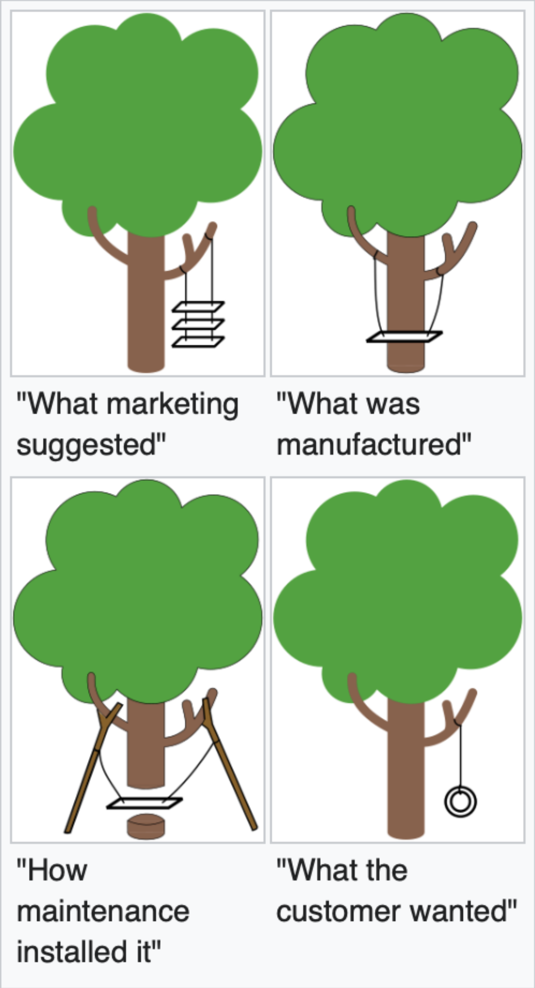
Requirements Analysis

User Stories

Testing Conditions of Satisfaction

Why care about requirements analysis?

Mostly? Long, painful experience



"What marketing suggested"

"What was manufactured"

"How maintenance installed it"

"What the customer wanted"

https://en.wikipedia.org/wiki/Tree_swing_cartoon

Why care about requirements analysis?

Essential for managing risk

Problems of understanding

Do users know what they want?

Do users know what we don’t know?

Do we know who are users even are?

Problems of scope

What are we building?

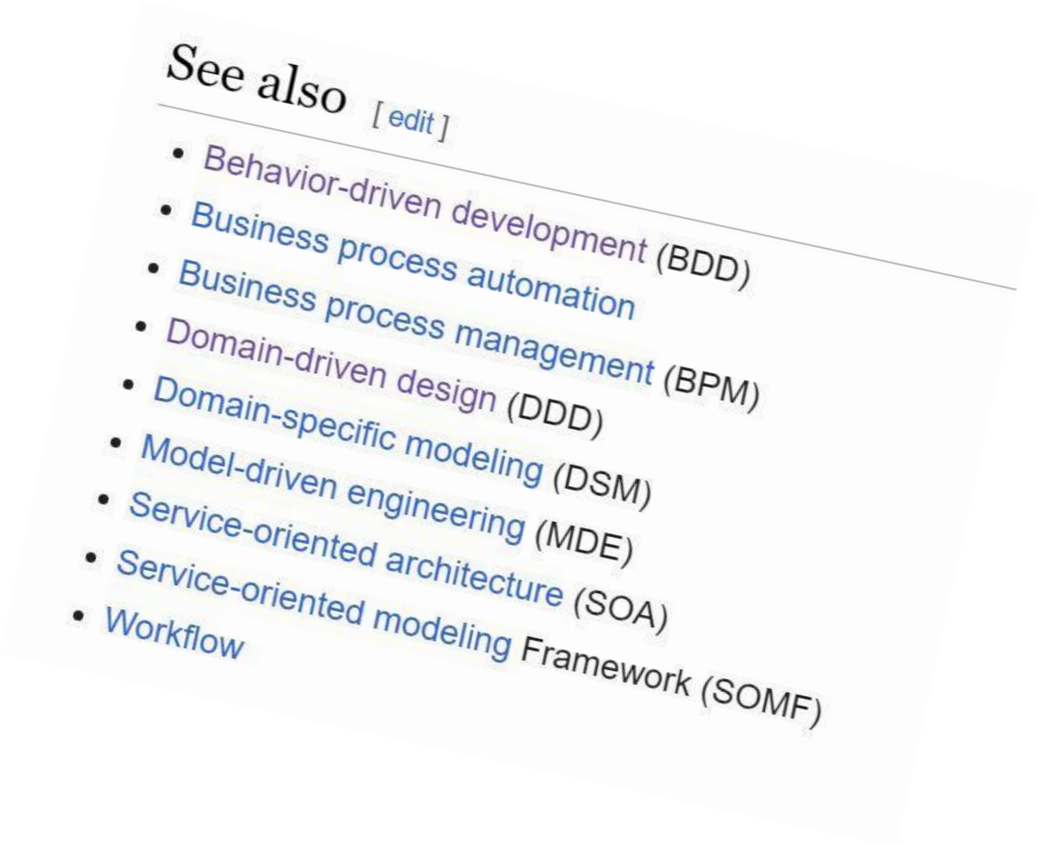
What non-functional quality attributes are included?

Problems of volatility

Changing requirements over time

How do we capture the requirements?

- There are many methodologies for this.
- Often described as x-Driven Design (for some x)
- They differ in scope & details, but they have many features in common.



See also [edit]

- Behavior-driven development (BDD)
- Business process automation
- Business process management (BPM)
- Domain-driven design (DDD)
- Domain-specific modeling (DSM)
- Model-driven engineering (MDE)
- Service-oriented architecture (SOA)
- Service-oriented modeling Framework (SOMF)
- Workflow

Common Elements

- Meet with stakeholders
- Develop a common language
- Collect desired system behaviors
- Document the desired behaviors
- Iterate, refine

- User stories are the least common denominator of most approaches

User stories in requirements analysis

- User stories (and conditions of satisfaction) define the minimum viable product
  - Requirements analysis determines the essential user stories
  - MVP has all the essential components of all essential user stories.
- Defining an MVP is important for your course project
  - User stories have equal footing — digging into requirements analysis for *prioritization* is outside of class scope
  - Half your code-based grade on the final project is based on the essential conditions of satisfaction
  - make your MVP too hard to complete (but don't make it too easy, either)

NOT user stories in requirements analysis

- User stories roughly describe the **functional requirements**
- The **non-functional requirements** are other properties that are also important to users and to other stakeholders?
  - How quickly can a transcript be retrieval? (Performance)
  - How many student transcripts can our system store? (Scalability)
  - How long did I spend on the phone with support to set up the software? (Usability)
  - After my system is setup, is the access controlled at all? (Security)
  - Are these any times when I can’t use this system? (Availability)
  - How expensive will it be to adapt the system to new requirements? (Maintainability)

Example:

- “The system should be able to service at least 200 simultaneous clients with less than 300ms latency, while costing less than $500/month on AWS.”

Other non-functional requirements

- Accessibility*
- Availability
- Capacity
- Efficiency
- Performance
- Privacy
- Response Time
- Security
- Supportability
- Usability

**We had a good conversation about this in class! Is it a “role” to be colorblind, or to be unable to use a pointing device? There are kind of two ways to answer:*

- *no, “colorblind user” is not a good role for the same reason “user” is not good; colorblind accessibility or screen-reader-friendliness is properly seen as a non-functional requirement*
- *yes, this is a role, and we need to think about roles at the intersection roles, like of ”colorblind x college administrator”*

I (Rob the instructor) am taking a relatively strict and prescriptivist stance on user stories in this class, and in class I said I was drawn to the first answer. Non-functional requirements can still be very important!

*However, I want to be clear that group projects **can** have roles in their user stories that involve , and I encourage you to do this if you have interest.*

Still more non-functional requirements

- Qualities that reflect the evolution of the system
  - Testability
  - Maintainability
  - Extensibility
  - Scalability

- Over-focusing on user stories and design requirements can be counterproductive in the long term! (Under-focusing on user stories and design can be terrible for a hackathon)
- Specifying the intended long-term maintainability of a system is part of requirements analysis — but beware changing requirements around longevity

Review

It’s the end of this lesson, so you should be able to:

- Explain the overall purposes of requirements analysis
- Recall the three major dimensions of risk in requirements analysis
- Explain the connection between requirements analysis and user stories
- Identify functional and non-functional requirements, and give examples of each