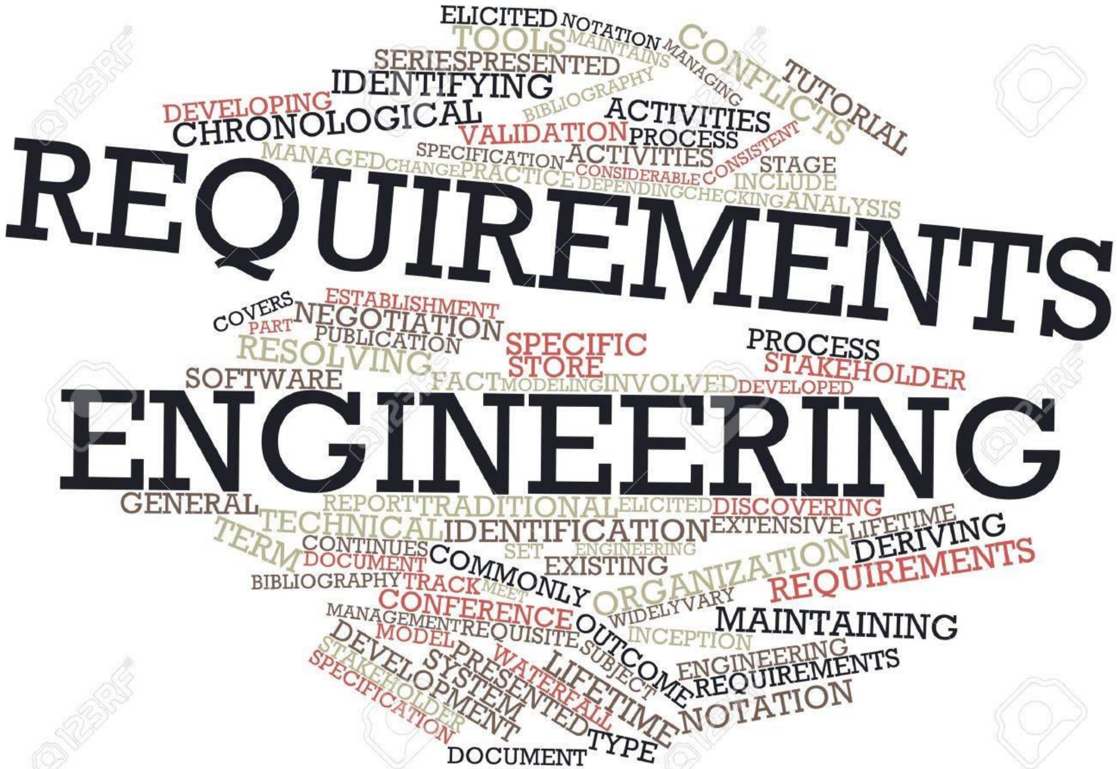


SWE 215: Software Requirements Engineering

Lecture 1

Why Requirements Engineering?

Course Topics

- **Why Requirements Engineering?**
- **Introduction to Requirements**
- **Requirements Engineering Methods**
- **System Vision, Context, and RE Framework**
- **Fundamentals of Goal Orientation**
- **Fundamentals of Scenarios**
- **Requirements Discovery**
- **User Stories and Agile Estimation**
- **Features Prioritization**
- **Requirements Negotiation**
- **Requirements Validation**
- **Fundamentals of Requirements Management**

Lecture Objectives

- ✓ Understand the importance of requirements engineering for the development of software intensive systems
- ✓ Embedding of requirements engineering into the organizational context
- ✓ Requirements Engineering in the development process

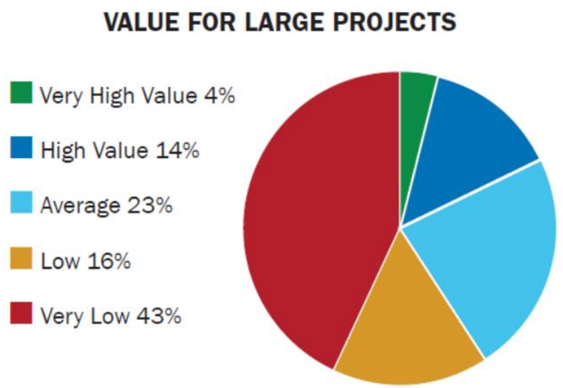
CHAOS REPORT 2015

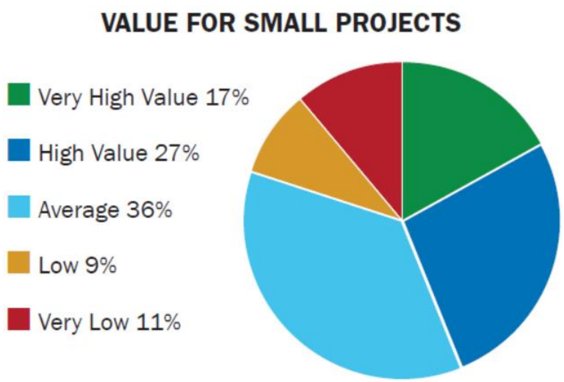
- Produced by the Standish Group
- Investigated software projects **success** and **failure**
  - **Successful**: project resolved within a reasonable estimated time (OnTime), stayed within budget (OnBudget), and delivered customer and user satisfaction (OnGoal, Value, Satisfaction)
  - **Challenged** (late, over budget (or both) with less than satisfied customers (did not meet their expectations)
  - **Failed** (Cancelled before it gets resolved or is resolved but never gets used)

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

CHAOS REPORT 2015: Project Size

- The larger the project, the less valuable the return rate. In many cases larger projects never return value to an organization.
- The faster the projects go into production the quicker the payback starts to accumulate.





	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	**100%**	**100%**	**100%**

CHAOS REPORT 2015: Project Complexity

- 38% of very easy projects were successful. Very complex projects have both the highest challenged (57%) and failure (28%) rates.
- Complexity is often caused by size, conflicting goals, and large budgets

CHAOS RESOLUTION BY COMPLEXITY

	SUCCESSFUL	CHALLENGED	FAILED
Very Complex	15%	57%	28%
Complex	18%	56%	26%
Average	28%	54%	18%
Easy	35%	49%	16%
Very Easy	38%	47%	15%

CHAOS REPORT 2015: Success Factors

FACTORS OF SUCCESS	POINTS	INVESTMENT
Executive Sponsorship	15	15%
Emotional Maturity	15	15%
User Involvement	15	15%
Optimization	15	15%
Skilled Resources	10	10%
Standard Architecture	8	8%
Agile Process	7	7%
Modest Execution	6	6%
Project Management Expertise	5	5%
Clear Business Objectives	4	4%

Unsuccessful (challenged and impaired) Project Causes

- One of the major causes of both cost and time overruns is **restarts**.
  - For every 100 projects that start, there are 94 restarts.
  - Some projects **can have several restarts**.

https://www.projectsmart.co.uk/white-papers/chaos-report.pdf

Project Challenged Factors

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

https://www.projectsmart.co.uk/white-papers/chaos-report.pdf

Project Impaired Factors

Project Impaired Factors	% of Responses
1. Incomplete Requirements	13.1%
2. Lack of User Involvement	12.4%
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements & Specifications	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%
Other	9.9%

https://www.projectsmart.co.uk/white-papers/chaos-report.pdf

Defects in Requirements

- The spectrum of failures ranges from failures with **minor consequences** to failure with **disastrous effects**.
- The reason for many of these failures lay in **insufficient requirements engineering**:
  - **Requirements gathering,**
  - **Requirements documentation,**
  - **Requirements management.**

Defects in Requirements: Mars Climate Orbiter

- In 1999, the Mars Climate Orbiter disappeared around Mars
- **Cost**: about $125M US
- **Problem caused by a misunderstanding between a team in Colorado and one in California**
- **One team used the metric system while the other used the English system for a key function**

Defects in Requirements: The London Ambulance Services (LAS) Failure

- LAS computer-aided dispatch system was designed to partially automate the manual **processing of emergency calls**.
- An emergency call should be answered by an employee of the LAS, who should ask for the location of the emergency.
- Based on this information, the system should determine the ambulances which are **close to the location of the emergency**.
- An ambulance **close** to the emergency location and **ready** for service should be **dispatched to the emergency**.

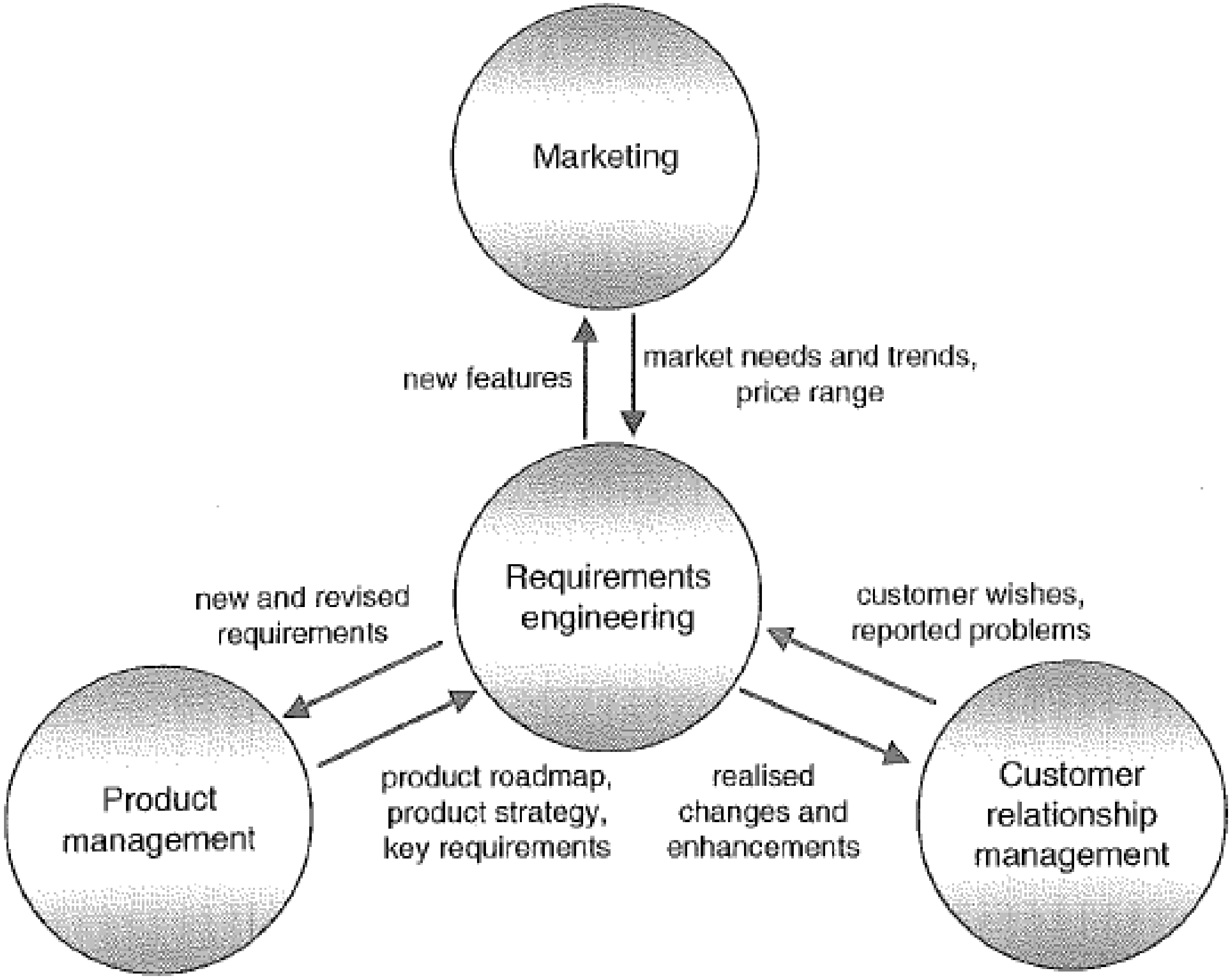
LAS Failure: Consequences of deficient Requirements

- The **RE process** for LAS system was **rather poor**:
- Ambulance crews were **insufficiently involved** in the RE process leading to:
  - **Inappropriate user interfaces** for the communication devices in the ambulances (**incorrect or insufficient information**). The crews had **difficulties in operating the devices correctly** and hence became frustrated in using the system.
  - The requirements did not account for the case in which **a crew might take a different ambulance** from the one assigned by the system.
  - Insufficient consideration of the communication network during RE, the existence of **radio black spots** was not considered in the requirements at all.

High Cost of Defects in Requirements Engineering

- Nearly **50% of the failures found in program source code** can be traced back to **requirements defects**.
- The **cost of removing defects in requirements increases the later the defects are detected during the development process.**
- If a requirement defect is detected during programming, the cost to fix the defect is approximately **20 times higher** than the effort required to detect and fix the defect during requirements engineering.
- If the requirements defect is not detected until acceptance testing, the effort required can be **up to 100 times greater**.

Embedding of Requirements Engineering in the Organizational Context



Embedding of Requirements Engineering in the Organizational Context

- RE is always **embedded within a specific organizational context.**
  - **Example**: if a dedicated **product management process** is established within an organization, requirements engineering has to interface with this process.
- Requirements engineering, for example, obtains product strategies, information about competing products, or key success requirements from the product management process.
- Conversely, requirements engineering passes new, innovative requirements to product management

Interrelations with other Development Activities