Introduction

Software Engineering 2

(3103313-1)

Amirkabir University of Technology

Fall 1399-1400

CHAOS Report 2015

	2011	2012	2013	2014	2015
Successful	29%	27%	31%	28%	29%
Challenged	49%	56%	50%	55%	52%
Failed	22%	17%	19%	17%	19%

Standish Group, CHAOS Report 2015

Why do projects fail so often?

Your experience!
Your thoughts!

Why do projects fail so often?

- **Unrealistic** or unarticulated project goals
- **Inaccurate estimates** of needed resources
- Badly defined system **requirements**
- Poor reporting of the project's status
- Unmanaged **risks**
- Poor **communication** among customers, developers, and users
- Use of **immature** technology
- Inability to handle the project's complexity
- **Sloppy development practices**
- Poor project management
- Stakeholder politics
- Commercial pressures

Factors of Success

Standish Group, CHAOS Report 2015

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%

Unfortunately ….

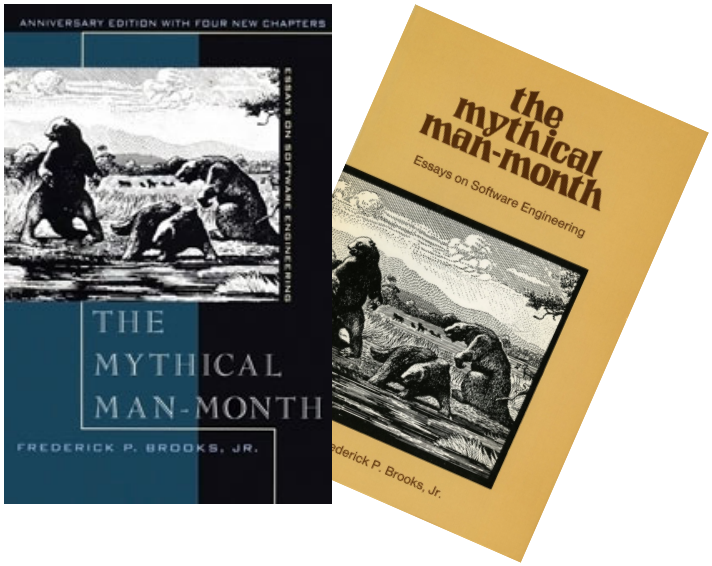
Big software failures tend to resemble the worst conceivable airplane crash, where the pilot was inexperienced but exceedingly rash, flew into an ice storm in an untested aircraft, and worked for an airline that gave lip service to safety while cutting back on training and maintenance. If you read the investigator's report afterward, you'd be shaking your head and asking,

"Wasn't such a crash inevitable?"

Charette, Robert N., "Why Software Fails," IEEE Spectrum Online, Sept. 2005.

break

The Mythical Man-Month: Essays on Software Engineering



- A book on **software engineering** and **project management** by **Fred Brooks** first published in 1975, with subsequent editions in 1982 and 1995.

- Brooks' observations based on his experiences at IBM while managing the development of **OS/360**.

https://www.greatersum.com/mythical-man-month/

- **The mythical man-month**
- No silver bullet
- The second-system effect
- The tendency towards irreducible number of errors
- …..

Software Process

Set of related activities that leads to the production of a software system.

Process Activities

many variations …

- Specification
- Design and Implementation
- Validation
- Evolution

- Communication
- Planning
- Modelling
- Construction
- Deployment
- +Umbrella Activities

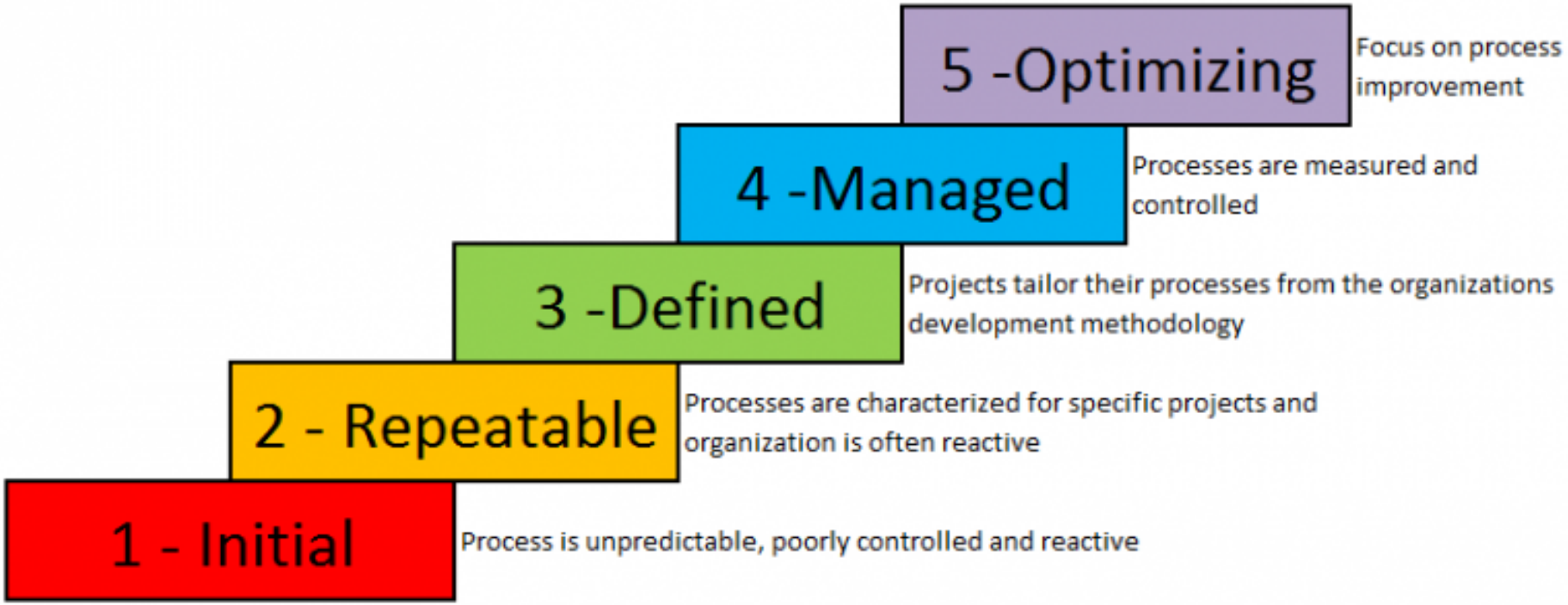
- Requirements Collection
- Analysis
- Design
- Implementation
- Test
- Maintenance

Selecting the Right Process (Model)

- Context/Project Characteristics
- Types of Application
- …

- Useful process (model) for
  - Unclear user requirements?
  - Unfamiliar technology?
  - Complex systems?
  - Reliable systems?
  - Short time schedule?
  - Schedule visibility?

CMM Process Management Model



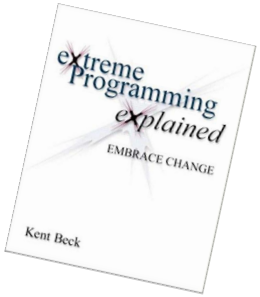
Agile Processes

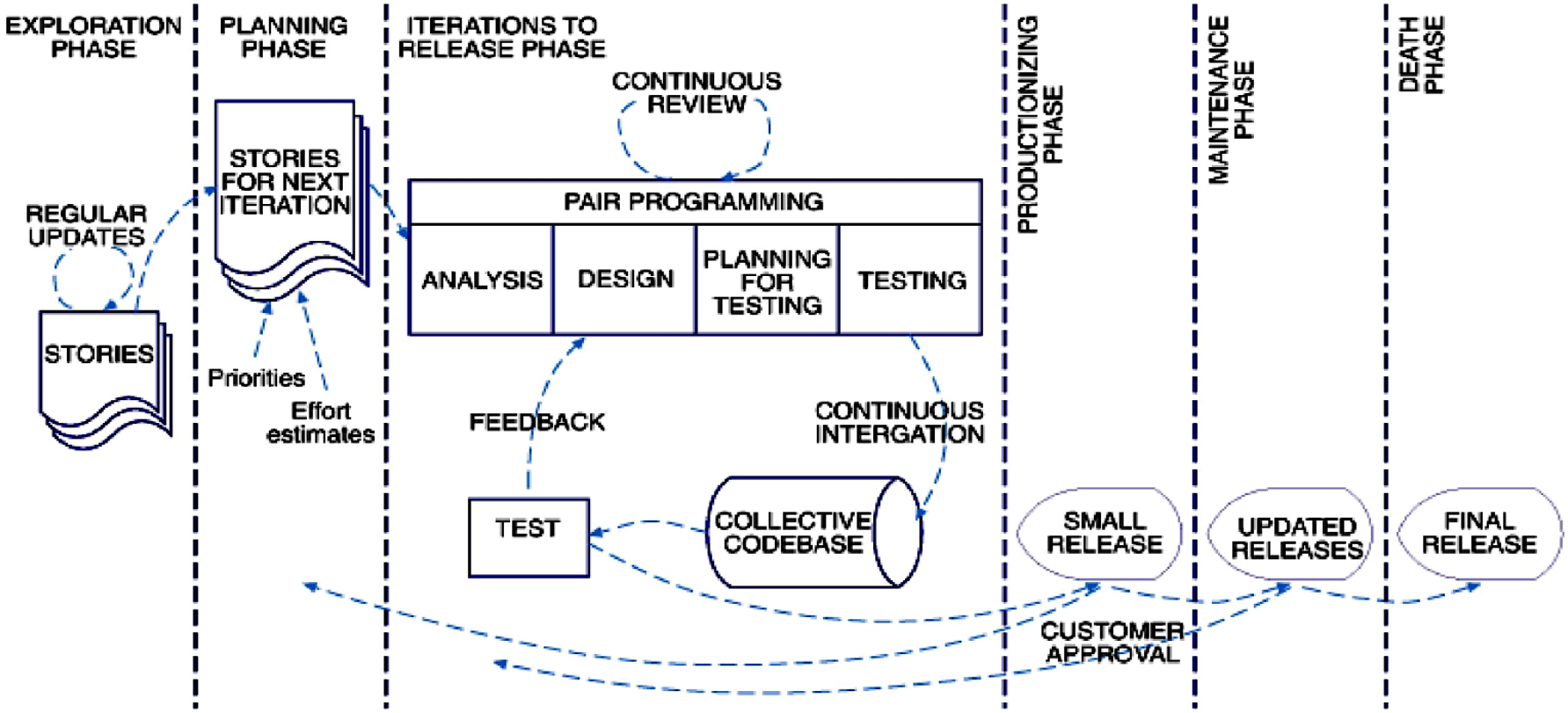
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

"That is, while there is value in the items on the right, we value the items on the left **more**"

e**X**treme **P**rogramming





Scrum

Scrum – Big Picture