Introduction to Software Maintenance (SB5-MAI)

Jan Corfixen Sørensen

University of Southern Denmark

September 1, 2017

Outline

SB5-MAI: Software Maintenance

Introduction

Paradigms

Software Life-Span Models

Summary

SB5-MAI: Software Maintenance

About SB5-MAI

- **Weight:** 5 ECTS / 0.083 full-time equivalent.
- **Prerequisites:**
  - Advanced skills in object-oriented programming (equivalent to 12.5 ECTS).
  - Software Engineering (process modeling, requirements, analysis, design, architecture, corresponding to 20 ECTS).
  - SB1-GOP-U1, SB2-VOP-U1, SB-ISE-U1, SB-GSE-U1 and SB-DES-U1, and project work.
- **Learning Outcome:** Maintenance of larger existing software projects.
- **Teaching language:** English or Danish.

SB5-MAI Exam

- The individual portfolio will constitute a part of the basis for examination and has to be approved before the final exam.
- Oral exam in combination with the student portfolio
- The exam is assessed using external examiner and the 7-point grading scale.

SB5-MAI Portfolio

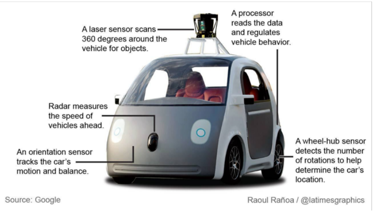
- Implement **new** JHotDraw features.
- **Individual** mandatory report.
- Two hand-ins (Deadlines will be announced soon).
- Has **to be approved** for going to the exam.
- **Rubric** normative feedback.

Introduction

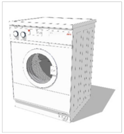
Motivation

- Internet in everything

Self driving cars


A laser sensor scans 360 degrees around the vehicle for objects.
A processor reads the data and regulates vehicle behavior.
Radar measures the speed of vehicles ahead.
An orientation sensor tracks the car's motion and balance.
A wheel-hub sensor detects the number of rotations to help determine the car's location.
Source: Google
Raoul Rañoa / @latimesgraphics

Washing Machines

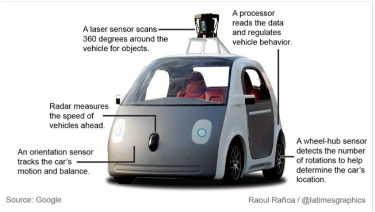


AI Watson Google Photos

Motivation

- Internet in everything
- Software is everywhere

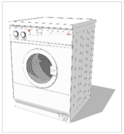
Self driving cars


A laser sensor scans 360 degrees around the vehicle for objects.
A processor reads the data and regulates vehicle behavior.
Radar measures the speed of vehicles ahead.
An orientation sensor tracks the car's motion and balance.
A wheel-hub sensor detects the number of rotations to help determine the car's location.

Source: Google

Raoul Rañoa / @latimesgraphics

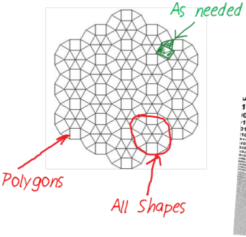
Washing Machines



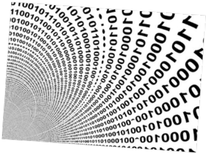
AI Watson Google Photos

Essential Properties

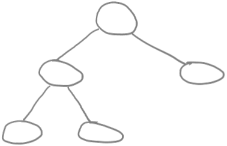
Complexity



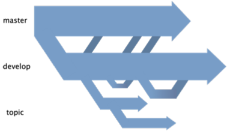
Invisibility



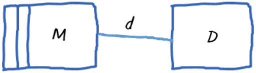
Visualization tools



Changeability



Conformity



Discontinuity

Accidental Properties

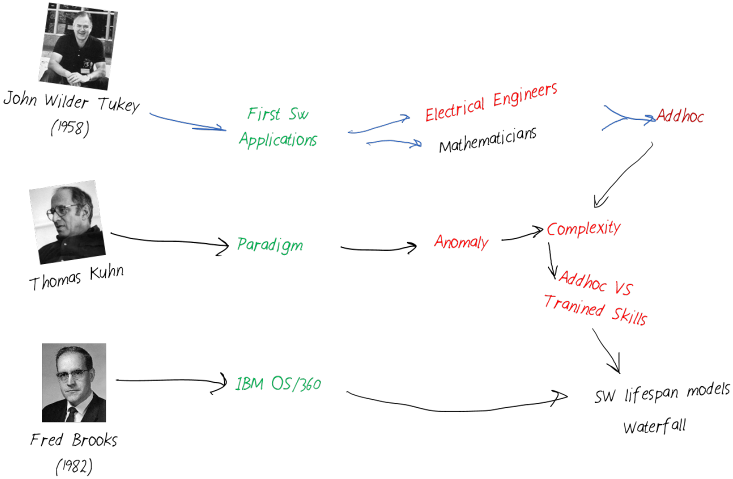
Technologies

GWT
Joomla!
Magento
AJAX
Microsoft Silverlight
Fl
Microsoft .NET
php
Java
iPhone
android
SharePoint
f
sitecore
ZF
ooVoo

Roles of Software

Operating systems

Software Engineering History



John Wilder Tukey (1958)

First Sw Applications

Electrical Engineers

Mathematicians

Addhoc

Thomas Kuhn

Paradigm

Anomaly

Complexity

Addhoc VS Tranined Skills

Fred Brooks (1982)

IBM OS/360

SW lifespan models

Waterfall

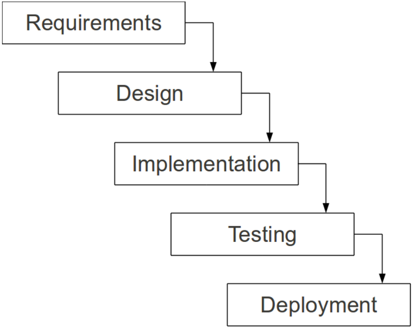
Paradigms

Ad hoc

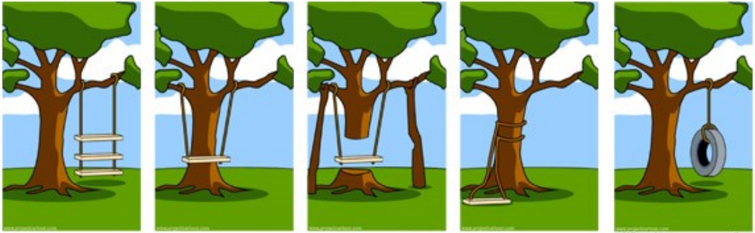
- Programmers recruited from the ranks of hardware engineers and mathematicians
- Software separated from the hardware in 1950's
- Software engineering defined as a field on it's own.

Software Waterfall

- Linear process
- Used in construction and manufacturing
- Intuitively appealing metaphor



Anomaly of Requirements volatility



How the customer explainened it.

How the project leader understood it.

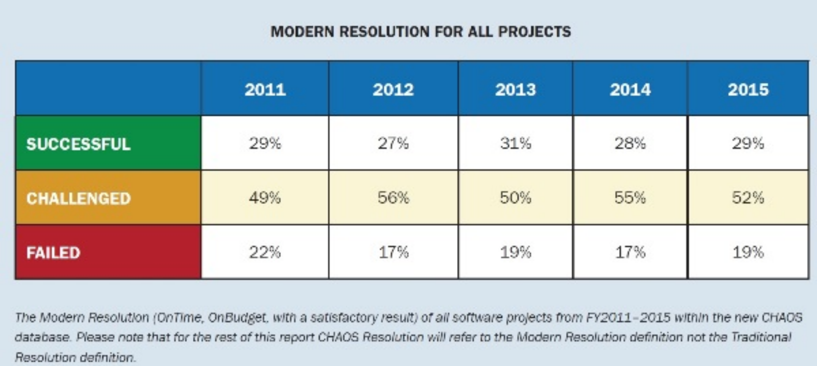
How the analyst designed it.

How the programmer wrote it.

What the customer really wanted.

- Requirements for IT change at a rate 2 to 3 % per month (Casper-Jones).

Standish Group Anomaly



MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

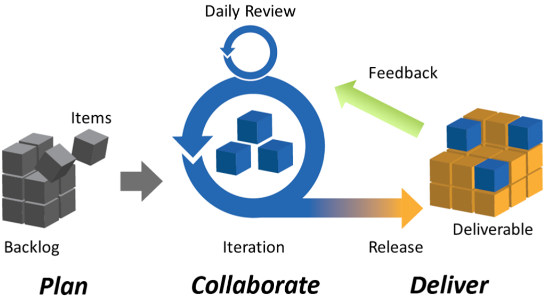
The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

- The CHAOS Reports published every year since 1994.
- 50,000 projects around the world.
- Based on new definition of success factors (on time, on budget with a satisfactory result).

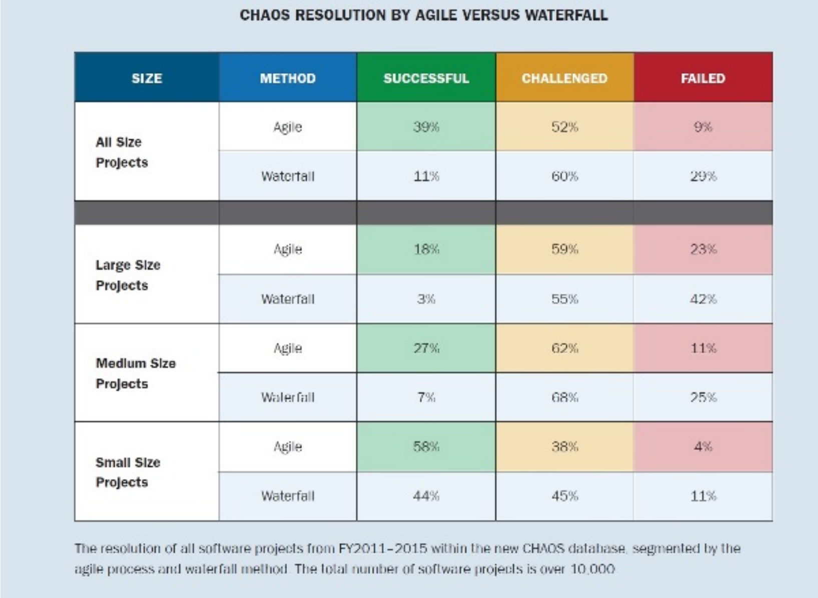
Agile Approach

Software Type	Monthly Rate of Requirements Change
Contract or outsource software	1.0 %
Information systems	1.5 %
System software	2.0 %
Military software	2.0 %
Commersial software	3.5 %

From „Strategies for Managing Requirements Creep", C. Jones, 1996, IEEE Computer



Agile VS Waterfall



CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

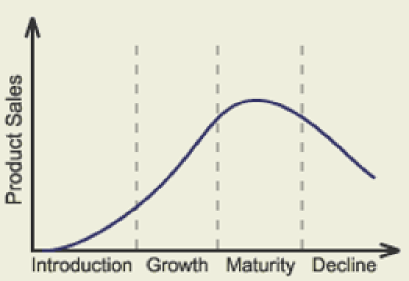
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

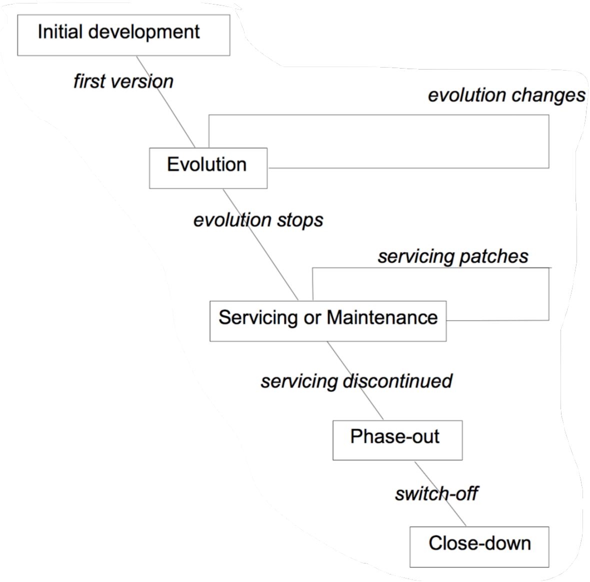
Software Life-Span Models

Life-Span Models

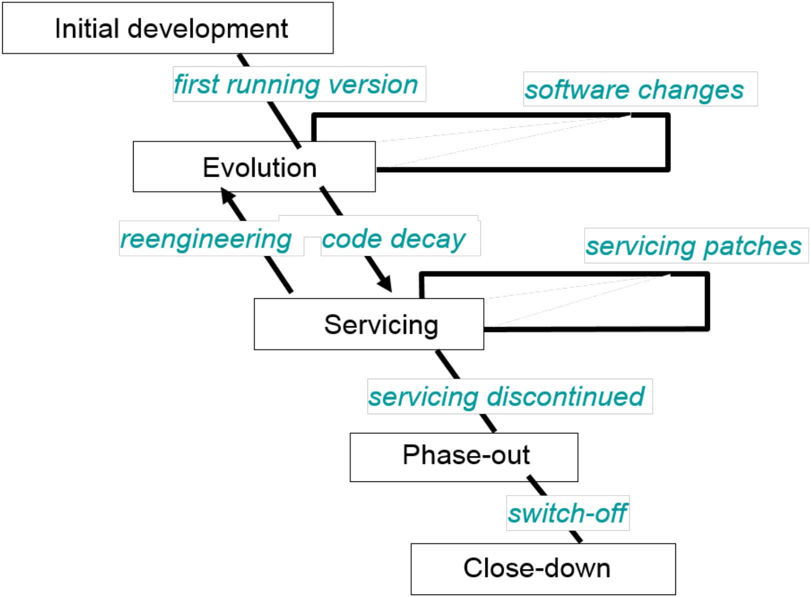
- Stages through which software goes, from conception to death.
- Stages may be very different.
- Stages are similar to the stages in the life span of other products.



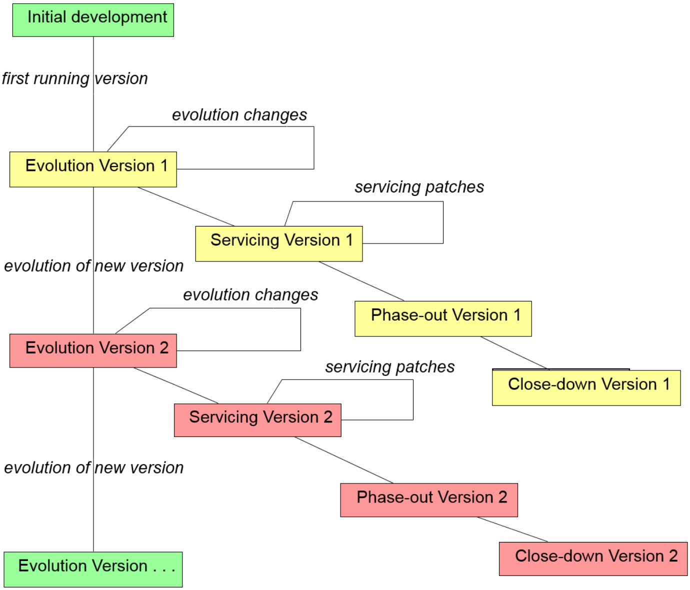
Staged Model



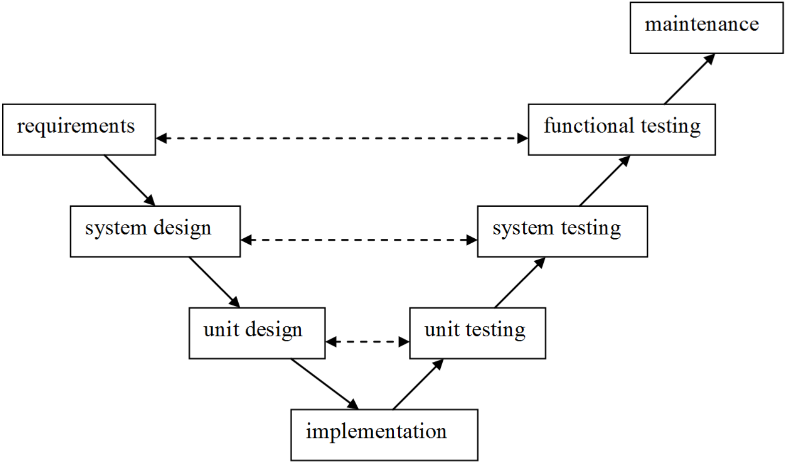
Reengineering



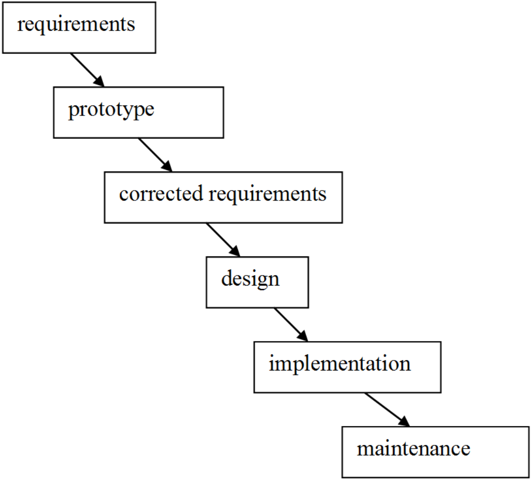
Versioned Staged Model



V-Model


maintenance
requirements
functional testing
system design
system testing
unit design
unit testing
implementation

Prototype Model



Summary

Software Engineering

- Sofware as a product since 1950s.
- Sofware engineers are faced with essential difficulties of complexity, invisibility, changeability, conformatity, and discontinuity.
- Three different paradigms:
  - Ad hoc
  - Waterfall
  - Iterative
- The different paradigms co-exists.

Software Life-Span Models

- Life-span models offer a simplified and comprehensive view of the entire software engineering discipline.
- Staged model
- Versioned staged model
- V-model
- Prototype