

Research vs Commercial Software Development

Nikki Appleby

Overview

- **My background**
- Software Development Processes
 - Commonalities
- Needs
- Commercial processes

My background

- Graduated BSc Computer Science 1989
- British Telecom (BT)
 - 250,000 employees
 - Ex-government department (Post Office)
 - 8th largest employer in world
 - Global customers
- Eastern Electricity
- AAPT
 - 11,000 employees

Platform

- Client
 - Windows v2
 - Visual Basic v4
 - MS Access v2 (prototyping)
- Server
 - Oracle 7
 - DBA

Progression

- Business Analyst
- Data and migration analyst
- Systems architect

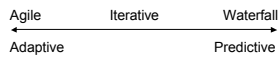
- Bioinformatics PhD
- Australian Centre for Plant Functional Genomics (ACPFPG)
 - Barley & Wheat

Overview

- My background
- **Software Development Processes**
 - Commonalities
- Needs
- Commercial processes

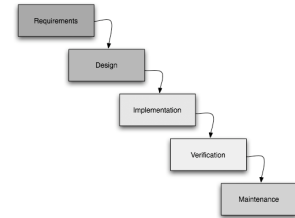
Development Processes

- Waterfall
- Iterative
- Agile



Waterfall Model

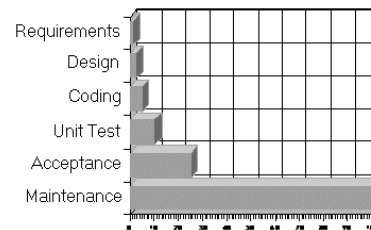
- 1970 - originally proposed by W. W. Royce
- Not as the author suggested
 - Was proposed to be iterative
- 2006 – most widely used



Waterfall Features

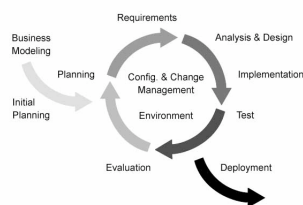
- Plan for future-proofing and reusability
 - Increases **cost**
- Costs set at start of project - **inflexible**
- Starts with benefits analysis
- **Slow**
 - 1-2 years
- Expensive

Big Design Up Front



Iterative Model

- Example
 - Rational Unified Process (RUP)



Iterative Features

- Faster
 - 3-6 months
- Perceived as **expensive**
- Has to be **adopted** and supported as main strategy
 - Difficult to use pieces in isolation

Agile Model

- Rapid Application Development / Prototyping
- Examples
 - Dynamic Systems Development Method (DSDM)
 - Consortium formed 1994
 - Extreme Programming (XP)
 - Refined 1996-1999
 - Kent Beck, Ward Cunningham, Ron Jeffries

Agile Features

- Fast
 - 1-3 weeks
- Focus on providing simplest solution leads to much **refactoring** later
 - Yet an aim is to lower the cost of change
- Still have to **design** to avoid conflicts of dependencies

Overview

- My background
- Software Development Processes
 - **Commonalities**
- Needs
- Commercial processes

Commonalities

- Requirements (Agile: “listening”)
 - Where you are going
- Design
 - How to get there without bumping into problems
- Testing
 - How you know when you have got there
 - Test that it does what you coded it to do
 - Test that it does what you wanted it to do
 - Against Requirements captured

Overview

- My background
- Software Development Processes
 - Commonalities
- **Needs**
- Commercial processes

Research Needs

- Not necessarily a clearly defined goal
 - Trial and error
 - Series of steps
 - Direction of next step not determined until prior step has been completed
- Get stuff done
 - Product or number of products
 - Fast
- Small team/solo project

Can lead to Ad-hoc-ery

- Research oriented
 - Perhaps no long-term strategy
- Object/process reuse
 - Faster to rewrite

Institute Needs

- Management of (teams of) researchers
- Coordination of strategy
- Cost management (maximum ROI)
 - Product or number of products
 - Quality
- Reusability
 - Products
 - Techniques
 - Data

Life Sciences

- Data generation
 - Maintenance and integration of laboratory systems
- Data management
 - Storage; security; accessibility
- Data analysis
 - Accessibility
 - Availability of software tools

Overview

- My background
- Software Development Processes
 - Commonalities
- Needs
- **Commercial processes**

Communication

- Data
 - Data documentation
 - Entity Relationship Diagrams (ERDs)
 - Data Dictionaries/Common Vocabularies
 - Ontologies
 - Standard data capture processes
- Architecture documentation
 - Installation instructions
- Domain experts

Development platform

- Common development platform
 - LAMP
 - Linux
 - Apache
 - mySQL
 - PHP / Perl / Python
 - Integrated Development Environment (IDE)

Deployment platform

- LAMP
- Windows
- Internet browsers
 - Which ones?

Software Quality



Kwaliteit van Softwareproducten - Ervaringen met een kwaliteitsmodel
 ("Quality of software products - Experiences with a quality model", in Dutch).
<http://www.ser.nl/quint-book/index.htm>

Review

- Did you do it the most efficient way
- Commercial
 - Cost effectiveness
 - Maximum benefit
 - Minimum cost \$\$\$
- Research
 - Time effectiveness
 - Maximum benefit
 - Minimum cost (resource/time)

Dangers

- "Optimism is an occupational hazard of programming."
 - Kent Beck
 - Time to develop
 - Features included
- Research also optimistic
- Manage both at same time?
 - Employ software development and quality processes

Requirements

