Software Processes

CMPT 276
© Dr. B. Fraser
Based on slides from Software Engineering 9th ed, Sommerville ch2
1) What activities are part of software development
2) What are software process models?
Process Activities
The software process

• Software Process:
  –

• All software processes involve:
  – Specification – what will the system do?
  – Design & implementation – how will it do this?..
  – Validation – does it do what the customer wants?
  – Evolution – change system to meet customer's changing needs.

• A software process model is..
Software Specification

Software specification: establishing what services are required and...

- Is it technically and financially feasible to build the system?
- What do the system stakeholders require or expect?
- Use gathered information to write a requirements document.
- Check the validity of the requirements

Flowchart:
- Feasibility study
  - Feasibility report
- Requirements elicitation and analysis
- Requirements specification
- System models
- User and system requirements
- Requirements document
Software design and implementation

- Process to convert system specification into an executable system.

<table>
<thead>
<tr>
<th>Design Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural Design</td>
<td>Identify overall structure of the system &amp; principle components:..</td>
</tr>
<tr>
<td>UI design</td>
<td>Layout initial ideas for user interface (UI).</td>
</tr>
<tr>
<td>Component design</td>
<td>Design each system component</td>
</tr>
<tr>
<td>Database design</td>
<td>Design the system's data structures and database</td>
</tr>
</tbody>
</table>
Software validation

• Validation
  - checks the system conforms to its..

• Involves testing
  - Create test cases which ensure system behaves correctly for some component/feature.
  - Best if using real-world data

• Can Involve Formal Verification
  - ..
  - Hard in practice; often restricted to critical components of life-critical components.
Testing Stages

Component testing

- Individual functions or objects are...
- May test coherent groupings of objects.

System testing

- Testing of system...
- Testing of emergent properties is particularly important.

Acceptance testing

- Testing with...
- to check that system meets customer’s needs.
Software evolution

- Software is inherently flexible and can change.
- Software must change to meet new business needs
  - Most of a project's time and cost associated with...

- Programming stereotype is:
  - development is creative and interesting, but
  - maintenance is dull.

  - This is increasingly irrelevant as most..

  - Line between old and new is blurring.
So, what's the process to develop software?

Software Processes
Software processes

• Describe each process by:
  – such as designing how data is stored, or the user interface, etc
  –

• All processes involve the four basic activities
  – specification, development, validation and evolution.

• 2 Big Questions
  – Done up front? Or as you go?
  – Done at the end? Or multiple times?
(Planning) Paradigms

• Plan-driven processes:
  -
  -
  - Also called Big Design Up Front (BDUF).

• Agile processes:
  -
  - Easier to change the process to reflect changing customer requirements.

• Most practical processes include elements of both plan-driven and agile approaches.
  -
Delivery

• Single Delivery (at end)
  – Software only delivered to customer..

• Incremental Delivery
  – Customer is given.. of the software throughout development.
### High-level View of Software Processes

**Single Delivery** | **Incremental Delivery**
--- | ---
Plan Driven (BDUF) | Plan Driven Incremental Model, Spiral Model
Evolutionary Planning |  

Describe what a course assignment would look like for each of these 4 possibilities.
Software process models

- The waterfall model
  - Plan-driven model – Separate and distinct phases of specification and development.

- Incremental development
  - Specification, development and validation are.

- Agile
  - Lightweight process to adapt to changing requirements.

- Most large systems developed using a process that incorporates elements from multiple models.
Waterfall model phases

- Requirements definition
- System and software design
- Implementation and unit testing
- Integration and system testing
- Operation and maintenance
Waterfall model problems

- Must complete phase N before starting phase N+1.
- Waterfall model is (somewhat) appropriate when:
  - Few business systems have stable requirements.
- Plan-driven nature of the waterfall model helps:
  - However waterfall is so rigid it is virtually never used as a full methodology.

“Walking on water and developing software from a specification are easy if..

-- Edward Berard (1993)
Incremental development

- Waterfall model delivers full system to user..
- Incremental development delivers..
Incremental and its benefits

- Incremental development usable by either paradigm
  - Plan Driven Models:
    Functionality of increments are..
  - Agile Models:
    Functionality of early increments are planned, later increments driven by...

- Reduced cost from changing customer requirements:
  - Not as much..

- Quick delivery of useful software.
  - Easier to get customer feedback on working software rather than paper designs.
  - Customer uses and gains value from the software earlier than with a single end delivery process.
Incremental development problems

• Code Rot:
  - Incorporating code changes becomes increasingly difficult and costly.
  - Time and money must be spent refactoring to improve the software.
Refactoring

- Refactoring
- Refactoring Examples
  - rename a poorly named variable
  - split huge function into smaller ones,
  - improve OOD (object oriented design)
  - fixing parts of the code which have..
Agile

- Agile methodologies are lightweight: they try to..
  - Ex: Only as much documentation and planning as needed.

- Develop application in short iterations
  - ~1-3 weeks long
  - .. at start of each iteration.
  - .. at end of each iteration.

- Very common in industry
  - Whole slide-deck on it soon!
Summary

- Software processes are the activities involved in producing a software system.
  - Requirements engineering: develop the specification.
  - Design and implementation: transform requirements specification into an executable software system.
  - Software validation: check the system conforms to its specification and meets the needs of its users.
  - Software evolution: change existing software systems to meet new requirements.

- Process models describe a sequence of activities: ‘waterfall’ model, incremental development, and agile development.