

Coping with Change and Risk

CMPT 276

© Dr. B. Fraser

Based on slides from Software Engineering 9th ed, Sommerville.

Topics

- How can software projects manage change?
 - What is prototyping?
 - What is incremental development?

Coping with change

- Change is inevitable in all large software projects:
 - Business changes lead to new (or changed) system requirements.
 - open up new possibilities.
- Cost of change =
 - Cost of reworking completed work
(re-analyzing requirements, design, re-coding)
 - +
 - Cost of..

Change avoidance with
(Throwaway)
Software Prototyping

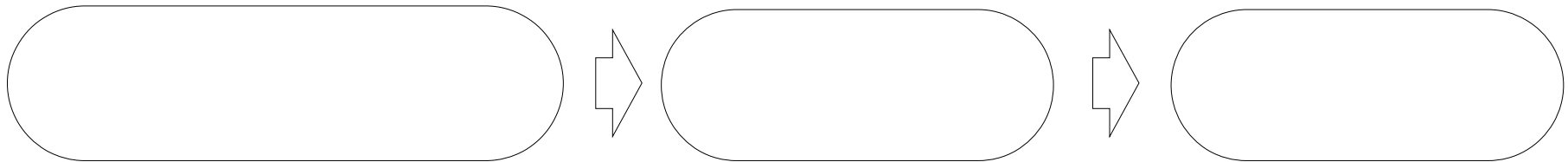
Throwaway Software Prototyping

- Prototype:
 - used to try out options.
- "Throw-away" code:
 - Prototypes could ignore things like code quality, error-handling, or testability.
 - Built to answer a specific question, not to see if the whole system will work.

Software prototyping

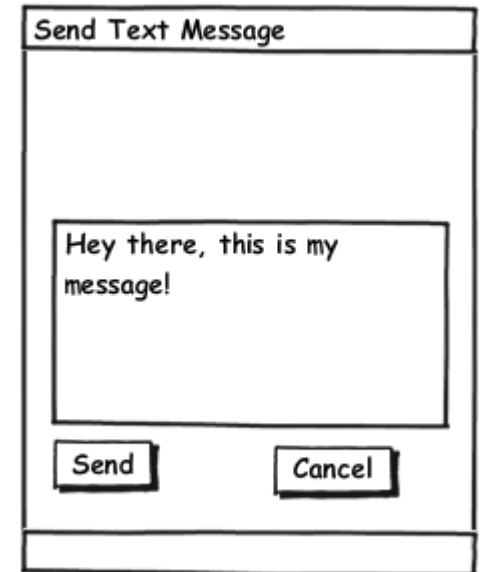
- A prototype can be used in:
 - ..
to help with requirements elicitation and validation;
 - ..
to explore options;
 - For example, a paper prototype of the UI.

Prototyping Process:



Benefits of prototyping

- Benefits of Prototyping:
 - Improved system usability
 - A closer match to users' real needs
 - Improved design quality
 - Improved maintainability
 - Reduced development effort

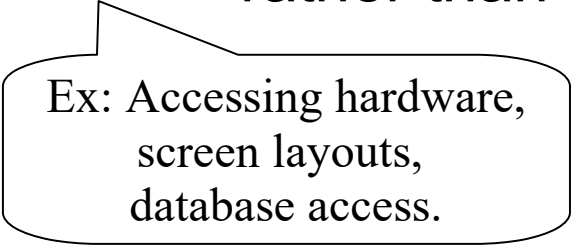


created with Balsamiq Mockups -

Prototype development

- ..
- Focus on poorly understood areas of the product;
- Error checking and recovery may be omitted;
- Focus on requirements.

rather than

A callout box with a pointer pointing to the text 'rather than'. It contains the text: 'Ex: Accessing hardware, screen layouts, database access.'

Ex: Accessing hardware,
screen layouts,
database access.

A callout box with a pointer pointing to the text 'rather than'. It contains the text: 'Ex: Security, performance, etc.'

Ex: Security,
performance, etc.

● Prototypes..

not a good basis for a production system:

- Likely below software quality standards.
- Normally undocumented;
- Degraded structure from rapid change (no refactoring)
- Hard to tune it to meet non-functional requirements.

Change tolerance with
Incremental Delivery

Incremental delivery

- Development and delivery are
 - ..
 - Each increment delivers some required functionality
- Prioritized user's requirements
 - Highest priority ones included in early increments.
 - Customer evaluates product at end of each increment
- Requirement changes
 - Once the development of an increment is started,
 - ..
 - Backlog's user stories continue to evolve; team will commit to these in future iterations

Customer value from incremental delivery

- Incremental delivery gives customer early benefits

Benefits & Drawbacks

What is a difference between an early increment and a prototype?

- Benefits include:
 - New functionality delivered with each increment so system functionality is available earlier.
 - Early increments act.. to help elicit requirements for later increments.
 - Lower risk of overall project failure.
 - Highest priority requirements implemented first and..
- Limitation of incremental delivery: Contracts
 - Complete system specification can be needed as part of the..

Summary

- Processes should cope with change.
 - Change avoidance:
 - Throwaway prototyping helps avoid poor decisions on requirements and design.
 - Change tolerance:
 - Iterative development and delivery allows changes without disrupting whole system.