SWINBURNE UNIVERSITY OF TECHNOLOGY

SWE30010
Development Project 2: Design, Planning and Management

Lecture 1

SDLC

Software Development Lifecycle (SDLC)

- The different stages developing software-intensive system

Software Development Lifecycle Model

- A model of the stages in SDLC
  - “Set of activities and their relationships to each other to support the development of a software system” (Bruegge and Dutoit)
- May be prescriptive or descriptive
- May be activity-centred or entity-centred
  (cf procedural vs OO software development!)

Various SDLC models

- Waterfall
  - and its variants such as V, Throwaway prototyping
- Iterative Development
- Spiral
- The Unified process model
- Scrum – an agile model (non-traditional)

Note: There are others SDLC models as well.

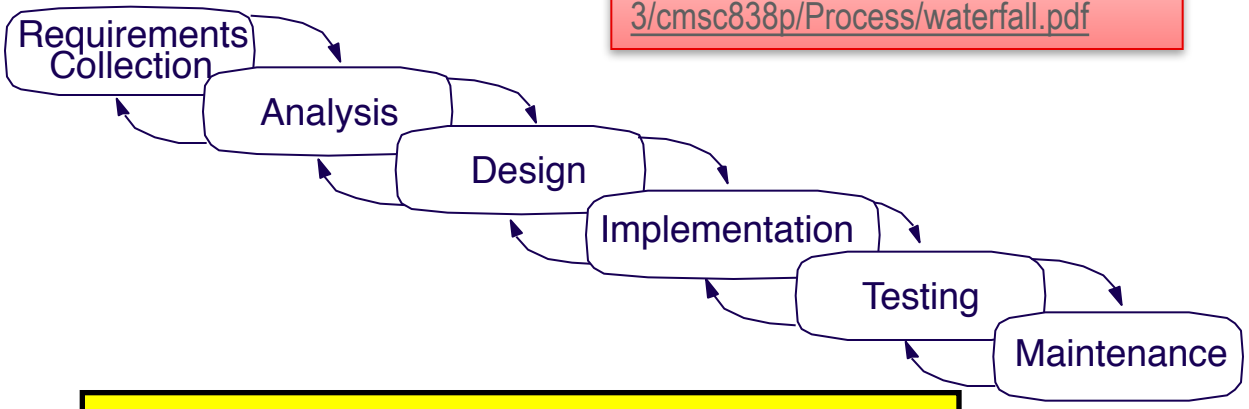
Macro Steps vs. Micro Steps

- Every SDLC model defines a process as an order of macro development steps
  - e.g.: Requirements analysis → design → coding → testing → …
- Within each macro step, there may be lots of "micro" steps
  - e.g.: in the coding macro step, coding of each "module" is a micro step, and within that, writing the code and unit testing it are sub-steps
- A macro step may involve a repetition of micro steps

Waterfall Model

The classical software lifecycle models the software development as a step-by-step "waterfall" between the various development phases.

Winston Royce:
http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf



You have seen this before, in SDP!!

The waterfall model is often problematic because:

- requirements must be *frozen early* in the life-cycle
- requirements are *validated late*

Problems with Waterfall Model

- No insight into how the transformation from one artefact to another takes place
- Requirements needed to be frozen at an early stage in the development (realistic for the kinds of military projects Royce was mainly concerned with)
- Too abstract to convey the complex process steps required to resolve the myriad of problems which arise at all stages of a software development.
- No recognition that software development is part science, part art.

☞ But: waterfall is quite adequate for *small, well-defined and well-scoped* problems!

Iterative Development

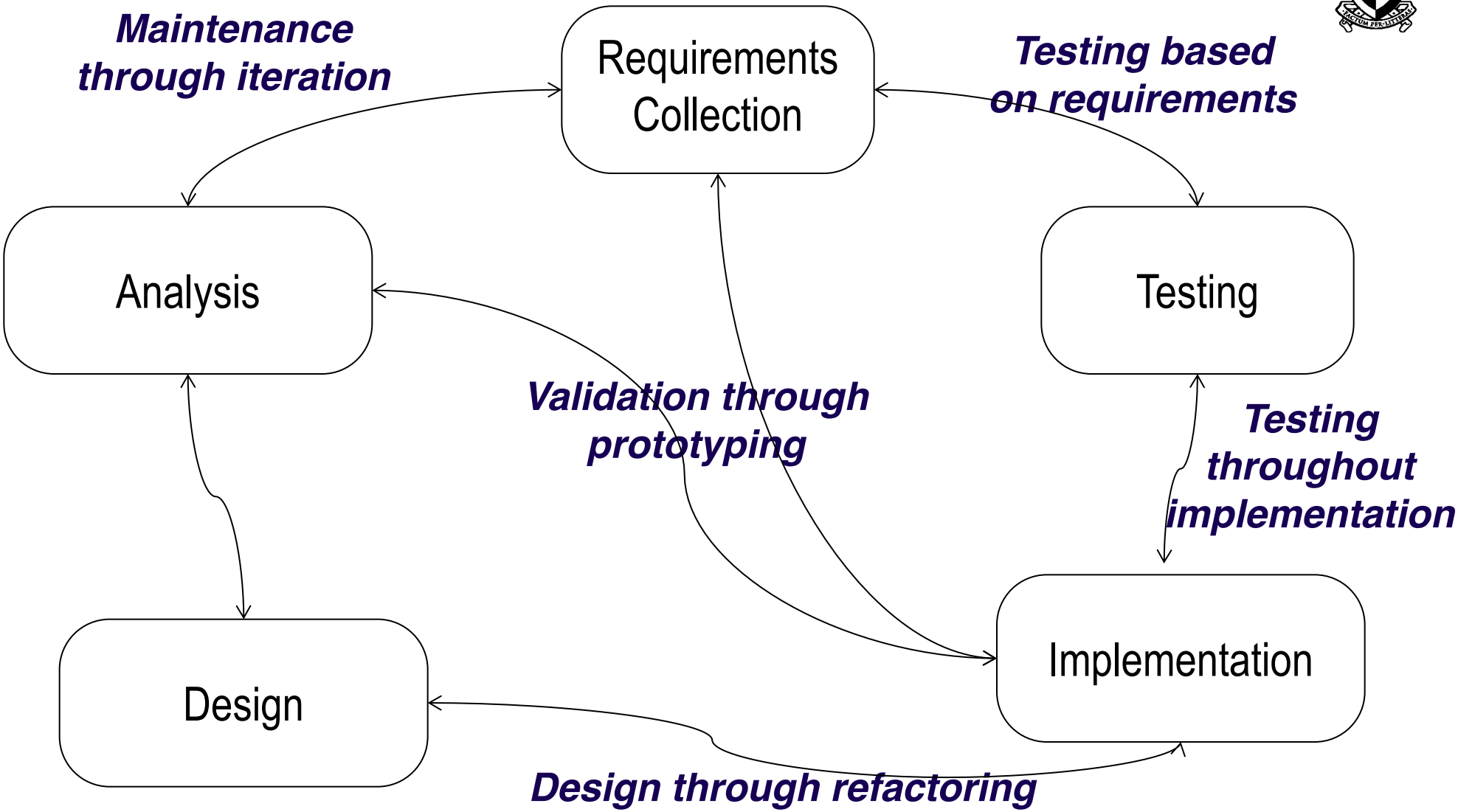
- In practice, development is often iterative, and activities progress in parallel (this has partly inspired recent approaches known as "*agile development*").
- Plan to *iterate* your analysis, design and implementation

  You will not get it right the first time, so *integrate*, *validate* and *test* as frequently as possible.

  "You should use iterative development only on projects that you want to succeed."

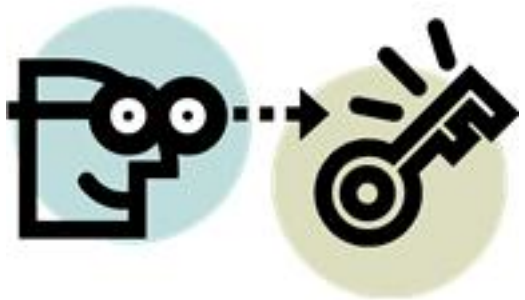
  — Martin Fowler, UML Distilled

Iterative Development (cont'd)



Iterative Development (cont.)

But……



Needs a very understanding and engaged client

Boehm's Spiral Lifecycle

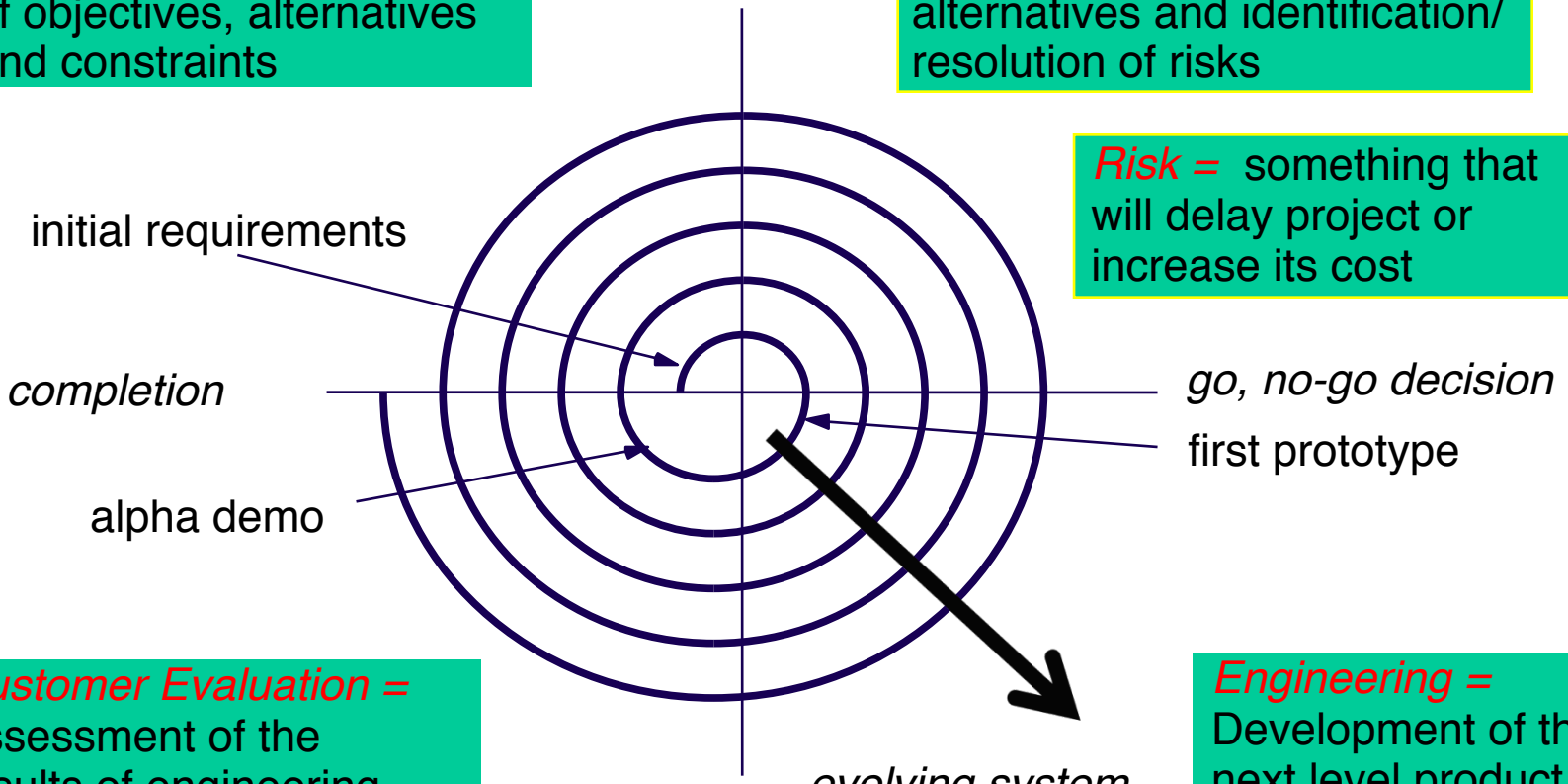
- A Management Process model with an emphasis on *analysing risks at regular stages* during development
- Risks are *consequences of inadequate information* and are resolved by initiating some actions to discover information which reduces uncertainty
  - ☐ interface risk -> develop prototype
  - ☐ feasibility risk -> buy information
- At each level of the spiral, any development model can be used
  - ☐ prototyping to resolve requirement risk
  - ☐ interface may be developed using reuse
  - ☐ conventional waterfall can be used too

Boehm's Spiral Lifecycle (cont.)

Planning = determination of objectives, alternatives and constraints

Risk Analysis = Analysis of alternatives and identification/ resolution of risks

Risk = something that will delay project or increase its cost



Customer Evaluation = Assessment of the results of engineering

Engineering = Development of the next level product

Source: Barry Boehm, *A Spiral Model of Software Development and Enhancement*, IEEE Comptuer, 21(5):61-72, May 1988

The Unified Process

4 phases:

- Inception
  - Idea refined, feasibility evaluated
- Elaboration
  - Project planning, requirements defined, resources allocated
- Construction
  - Development
- Transition
  - Installation, maintenance

Cycle

References: http://www.rational.com/products/rup

http://www.ootips.org/rup.html

Scrum – an Agile Development Method

See Lecture 1 Scrum

Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Choosing an appropriate SDLC

- Monthly sales reporting system
  - Software that analyzes sales data and prints the monthly sales report
- Flight simulation system
  - Software that simulates the flying of an airplane to train the pilot
- X-ray medical imaging system
  - Software that controls an X-ray machine to take image of human body tissues

What you should know!

- What is an SDLC model?
- How many SDLC models can you name and describe?