SE2: Introduction to Software Architecture

Reid Holmes

Lecture 2 - Thursday, September 15 2011.

Material and some slide content from:
- Emerson Murphy-Hill
- Software Architecture: Foundations, Theory, and Practice
- Essential Software Architecture
Course Objectives

By the end of the course you will be able to:

- propose and analyze software architectures
- explain the strengths and weaknesses of various architectural styles and design techniques
- communicate and rationalize architectural and design decisions
- ideate, justify, and implement software designs
- evaluate different architectures and designs
Project

Group organization

Project proposal

Next Friday’s tutorial
Group Organization
Project Proposal

- Due September 20 @ 0800
- See course web page for complete requirements.
Project Proposals

All SE2 Teams
RendezVous

Set Location
Set Image
Set Alarm
Start
Reset

Distance: 1.2km
Time: 15min
Level: MF

Show Picture
Finish
Project Title: To develop a Waterloo guide mobile application (multiple platforms) using GWT (K-W maps).

Group Name: LTZ
CS Advisor Appointment System
Team Legendary Apps
Event Chimp for Facebook
Road Trip
Friday Sept 23 Tutorial

- By the **end of Friday Sept 23** these should work:
  - Basic GWT project: [instructions]
  - Interesting GWT project: [instructions]
  - GWT Overview Video [link]
  - GWT v2 Overview [link]
  - Great starting tutorial [link]
  - Good overview slides [link]
- Some Resources:
  - GWT Widgets [link]
  - GWT Panels [link]
  - GWT Examples [link]
Architecture

- Architecture is:
  - All about communication.
  - What ‘parts’ are there?
  - How do the ‘parts’ fit together?

- Architecture is not:
  - About development.
  - About algorithms.
  - About data structures.
What is Software Architecture?

- The conceptual fabric that defines a system
  - All architecture is design but not all design is architecture.

- Architecture focuses on those aspects of a system that would be difficult to change once the system is built.

- Architectures capture three primary dimensions:
  - Structure
  - Communication
  - Nonfunctional requirements
Structural example
Non-functional requirements

- Technical constraints: restrictions made for technical reasons
- Business constraints: restrictions made for business reasons
- Quality attributes: e.g., the ‘ilities’
  - Scalability
  - Security
  - Performance
  - Maintainability
  - Evolvability
  - Reliability/Dependability
  - Deployability
“The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.
ANSI/IEEE 1471-200

“Architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution”
Eoin Woods

“Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.”
Philippe Krutchen

“The life of a software architect is long (and sometimes painful) succession of sub-optimal decisions made partly in the dark.”
WWW Example
Activity

- Architect a kitchen
  - What are the high-level requirements?
  - What systems are involved?
    - What properties do the systems have?
    - How do they interact?
  - How is the kitchen used?
Kitchen Architecture Discussion

- What are the components?
- What are their properties?
- What are their relationships?
- Present and critique design alternatives.
Upcoming Deadlines

Project proposal.
Pass/fail.
0800 Tuesday Sept 20.
Email rth.se2@gmail
Full description on course web page.