Appreciation for Software Architecture

SWEN-261
Introduction to Software Engineering
Department of Software Engineering
Rochester Institute of Technology
Why is architecture important?

- Martin Fowler in *Patterns of Enterprise Application Architecture*:
  
  "The highest-level breakdown of a system into its parts; the decisions that are hard to change; there are multiple architectures in a system; what is architecturally significant can change over a system's lifetime; and, in the end, architecture boils down to whatever the important stuff is."

- **Solution communication and consensus among stakeholders**

- **Earliest and most fundamental design analysis and decisions**
  - Directs and constrains remaining software development, deployment, and maintenance
  - Dictates structure of development organization
  - Enables early evolutionary prototyping
  - Enables more accurate cost and schedule estimation
Architecture is the highest-level design of a system.

**Component level**
- Idioms
- Application of design patterns

**Package level**
- Interface design
- Deployment planning

**Subsystem level**
- Separation of system into subsystems, logical layers
- Application of enterprise level patterns

**System level**
- High-level technology analysis for solution space
- System design based on high-level requirements
- Risk identification and mitigation
Requirements affect the system architecture

- Non-functional requirements (NFRs) and constraints lead to a logical and physical architecture.

- Operational NFRs:
  - *Scalability*
  - *Availability* ...

- Developmental NFRs:
  - *Testability*
  - *Portability* ...

- Constraints:
  - *Pre-chosen system components (eg, database)*
  - *Pre-chosen frameworks* ...
Architecture is guided by principles and patterns

- Manage risk
- Build vs buy vs open source
- Separation of concerns

Architectural patterns are selected to satisfy NFRs

- **Failover and Load Balancing**
- **Model-View-Controller**
- **Tiers and Layers**
  - For example: UI, Application and Model
- **Java EE Patterns**

- No one architecture is right or wrong, just more or less useful for a given application. (attribution unknown)
- **Does it satisfy the NFRs?**
- **Ad-hoc architecture is not very useful.**
WebCheckers Inc hired a contract architect who produced the Vision document:

**Architecture**

The WebCheckers webapp will use a Java-based web server. The team will use the Spark web micro framework and the FreeMarker template engine to handle HTTP requests and generate HTML responses. The team has access to the standard Java v8 libraries and language features. The team may also include additional, third-party libraries approved by the Product Owner.

**Process**

The team will use the OpenUP method for planning activities across the life span of the project. The team will use the Scrum process for day-to-day operations.

First, the application must be on the web.

The architect has also identified several frameworks to use.
Build out the architecture in the Elaboration phase

- The architecturally-significant user stories are prioritized during the Elaboration.
- The development team is frequently guided or lead by an architect during this phase.
- The working increment at the end of Elaboration forms the starting point of the system architecture.
  - *There will be architectural additions over the lifetime of the system.*
  - *Avoid changing established architectural norms.*
The User interacts with the User Interface (UI) tier.

The UI tier interacts with the Application and Model tiers.

The Application tier holds logic that controls the flow of the application.

The Model tier holds the core domain (aka "business") logic.
Architecture must also consider layers

- **System**
  - **Frameworks**
  - **Platform**
  - **OS/Hardware**

  ![Diagram](630x2 to 717x82)

- **UI**
- **Application**
- **Model**
Let's start with a simple, desktop architecture
This is the architecture you will use in the web-based term project.

**Client UI**
- User
- HTML, CSS & JavaScript
- Any Browser
- Any OS/HW

**Server UI**
- Network Connection
- Spark & FreeMarker

**Application**
- Java Web server (Jetty)
- Any OS and HW

**Model**
- Any OS and HW