**HW 6 – CS 4321, Fall 2015**

|  |  |
| --- | --- |
| Name: | Tyler Cabutto |

**Questions – Lesson 7: Software Architecture (37 minutes)**

Watch videos, type answers (leave questions), print before class and turn-in in class.

1. When building a software system you go through a process of making many design decisions many of which don’t affect the success of your system. Some, however, are the equivalent of load baring walls in a building. (Supply the missing phrase)
2. What is architectural erosion?

* Adding a feature or changing a piece of the user interface, porting to a mobile platform, etc. and end up with a software system that may be a distant relative to your original system. It is a mutant in many ways. This is how flaws can begin to creep into your system. This comes in to play where you have to figure out how to keep it alive and going.

1. What four non-functional qualities of a system can degrade with architectural erosion?

* 1. Security.
* 2. Reliability.
* 3. Usability.
* 4. Maintainability.

1. Define prescriptive architecture and descriptive architecture.

* **Prescriptive Architecture:** Captures the design decisions made prior to the system’s construction.
* **Descriptive Architecture:** Describes how the system has actually been built.

1. When a system evolves what *should* happen first?

* Ideally its prescriptive architecture should be modified first.

1. Why do prescriptive and descriptive architectures tend to diverge as systems evolve?

* 1. Plain sloppiness
* 2. Short deadlines.
* 3. Lack of documented prescriptive architecture.
* 4. Etc.

1. I didn’t understand what the author was saying the difference between architectural drift and erosion are? Can you explain that to me?

* Architectural drift becomes too complex by adding things to it that are deemed unnecessary, and architectural erosion is having a poor architecture, meaning that in the future it will cause the system to erode (issues with security usability, etc.).

1. What is architectural recovery?

* Determining a software architecture from an implementation and fixing it.

1. Define cohesion and coupling.

* **Cohesion**: The degree of which components belong together. Having high cohesion is good.
* **Coupling**: Having connections with other components. Having high coupling is not good

1. Provide an example of the pipe-and-filter architectural style that is **not** mentioned in the lecture (e.g. google it)

* Compilers: The consecutive filters perform lexical analysis, parsing, semantic analysis, and code generation.

[**Source**](http://www.dossier-andreas.net/software_architecture/pipe_and_filter.html)**: http://www.dossier-andreas.net/software\_architecture/pipe\_and\_filter.html**

1. Use google to find some basic information on REST. Summarize what REST is.

* REST is the software architectural style used by the World Wide Web. You’re interacting with resources and your interactions are specified through URL’s that addresses resources. It relies on a stateless, client-server, cacheable communications protocol and in almost all cases uses HTTP protocol.

1. What does the author say is wrong with Napster’s architecture?

* The peer and content directory can likely cause problems whenever the number of peers grows too large. The reason being is because the number of requests to the directory will become too much and keep up with it.

1. How does (did) skype address this shortcoming?

* They used super nodes which support high bandwidth that are not behind a firewall and runs skype regularly.

1. What did Microsoft do to replace the super-node idea discussed in the video? (e.g. google it)

* They supplied centralized Linux machines in order to be used as the super nodes rather than having user supplied super nodes.