

## CPSC-5200-01 Software Architecture and Design

### Mid-term

Please answer the following 5 questions and return to me by 8:00. There's a 6th question that can be used as a bonus. The overall exam is worth 20% of your quarterly grade, but not more than that (that is, answering the bonus question AND all of the others won't net you more than 20%). When you are finished with the exam you are free to leave. Please be sure to write your name at the top of each page.

This is a closed book exam. You may not use notes from class, my slides, or any of the other reading materials that I linked.

As we've discussed in class, I'm looking for you to think outside of the box on many of these questions. There's a reason why you won't find the *exact* answer to a question in the notes: it's because we haven't covered the exact scenario and you'll need to think about how you would address that scenario.

#### The questions

1. We can define a software architecture model as a collection of boxes and lines. But, if we do this we need to give semantics to those boxes and lines. What do they generally represent? (Hint: they do not represent services, or data access layers, or clients).
2. Technological decisions impact the economics of your software projects. Besides these, what other factors or decisions may have an impact and why do we care?
3. Is software architecture, and by extension the role of the software architect, compatible with Agile processes? In what ways might it be or not be compatible?
4. In his 2000 Ph.D thesis, Roy Fielding argues that REST is a logical evolution of existing architectural styles. How does he "create" the REST architectural style from seemingly nothing?
5. As a software architect you have a number of tools that you can lean on during the design process. However, one of those tools outweighs all of the others. What is that tool and what makes it so powerful?
6. (bonus) An architecture that addresses mainly how connectors and components are put together runs the risk of introducing an  $N^2$  problem. Solving that using a centralized approach introduces a potential single point-of-failure. How might one design a solution that mitigates these concerns?