RUP – the process

1. What is the most fundamental difference between the waterfall and iterative SE methodologies? Answer in one sentence!

Rup is risk driven, has low and high ceremony depending on project size, RUP has continuous integration

3. In which RUP phase can you start writing code?

Elaboration

4. In the RUP, you have a functional product very early on. How does the process enforce this?

Early integration

5. The RUP is risk-driven. What risk in minimized in Inception?

Project critical risks

6. What risk is minimized in the second RUP phase?

Large risks

7. So far we talked about the roles of the architect and the analyst. At which phase of the RUP does each of these roles usually have their most busy time?

Analyst spends the most time in elaboration and inception. Architects spend the most time in construction and elaboration

8. What is the SRS? Explain what its parts are and comment on each part’s part!

Use case document, supplemental specs, glossary, rules dictionary

9. See last question: what other documents are important for the Inception phase?

Vision document and business case

10. The RUP is also “use case-driven”. What does this actually mean?

Driven by how the software is used by its clients. The use cases are very important and a lot of time is spent on refining and detailing them.

11. Explain the steps that you go through (over and over again) in Construction!

Detail the use case and implement each one in order of most critical to least critical

Architecture

12. In the RUP, the notion of architecture is vital to the overall process. What is architecture?

Architecture is the foundation for the entire system. Like the operation system, it must be able to support the software being built on it

13. What is an executable architecture?

An architecture that your software can run on top of

14. Why is baselining an executable architecture so important for the RUP?

To find out if it will support your system

Analysis and Design

15. What is the “realization” of a use case?

When you think about the use cases that are needed and think of one that is critical and detail it’s functionality as much as possible

16. What are the three types of analysis classes? Describe their roles!

Boundary – translates, Control – Delegates work, Entities – Model key concepts

17. What are “mechanisms” in the RUP and why are they important?

They describe parts of the software that are needed to function, they are important for understanding how the software will meet the requirements of the project.

18. Let’s assume you have the analysis mechanism “persistence”. Give an example for a corresponding design mechanism and implementation mechanism!

You need a system like Persistent storage being the design mechanism, and MySQL being the implementation mechanism

19. Why do you need to think about key abstractions early on, i.e. before doing any design work?

Because the key abstractions lead to properly laying out the implementation classes and determining how each class uses the other systems