

ASSIGNMENTS WEEK 1

1. VIEWS AND VIEWPOINTS (20 PT)

- a) How do the viewpoints as defined by Rozanski & Woods and the RUP 4+1 views as defined by Kruchten relate to each other?
- b) The main difference between the definitions of architecture as given by Rozanski & Woods and the definition of IEEE is the clause “the principles guiding its design and evolution”.
 - Why do you think the IEEE included this in the definition?
 - Give an example of a system and two principles “guiding its design and evolution”.
- c) Chapter 5 of ISO/IEEE 42010 standard describes the contents of an architectural description. Compare this to the RUP SAD template. If there are any topics missing, is that a problem?

2 SOFTWARE ARCHITECTURE DOCUMENTS (20 PT)

These questions concern the example SAD's on Blackboard.

MOBILE AGENTS

- a) In the SAD *Mobile Agents*, paragraph 4.3, the choice for the agent framework is an important design decision. Write it down using the design decision format presented at the lectures.
- b) Which RUP view is missing in this document? Is this a problem?
- c) Give an UML deployment diagram to add to chapter 5.

GAIUS

- d) In *GAIUS Architecture Description* chapter 2 talks about “system roles”. Are these comparable to stakeholders or to use case actors?
- e) Explain the difference between stakeholders and use case actor. Give examples.
- f) To which RUP view does the information belong that is presented in 4.2.3?
- g) In 4.3.1 the Blackboard pattern is mentioned. This is an example of an *architectural pattern*. Explain in a few sentences what an architectural pattern is.
- h) Explain the Blackboard pattern.

3. REVERSE ENGINEERING OF A SOFTWARE ARCHITECTURE (60 PT)

This is the main exercise for this week. Make sure to start soon and give it enough time.

In this assignment you will document a software architecture according to the RUP 4+1 approach. You will try to unearth the structure of an undocumented system and experience the maintainability of an agile developed system.

The system used for this exercise is the end product of the Agile Software Development course (T4.1) created by another project team.

The setting is that you are working as developers at company XYZ. The company is considering accepting the maintenance of an externally developed system so it can extend it in the future. To judge whether this is wise you are asked to screen the code and report on it.

There are two stakeholders to consider. First there is the development/maintenance team, represented by team leader Bob. Can they maintain this system or would they be getting a big pile of problems? What do they need to know to be able to maintain the product smoothly?

The other stakeholder is the management, represented by the director Martin. Would it be interesting for him to get this product in his portfolio? What are the main features, characteristics, advantages and disadvantages of the product? How much time (money) will it cost to get it maintained?

- a) What are the **main quality attributes** the stakeholders are interested in? Give three quality attributes for each stakeholder.
- b) Give two other possible stakeholders we could have considered for this system. What would their **concerns** be?
- c) Study the system (source code and available documentation). Write a **SAD** based on the template at www.rupopmaat.nl. Make sure to include UML component diagram(s), package diagram(s) and deployment diagram(s).
- d) Identify three important **design decisions** that have been taken for the system. Where these design decisions documented by the original developers? What were the alternatives? What decision would you have made, with what rationale?
- e) How could the architecture of the system be **improved**? Give suggestions for improvement, at least one for every view.
- f) The standard views are probably too technical for Martin. Write a **management view**. You are allowed to make reasonable assumptions. (This exercise is about writing for a non-technical person and being able to imagine the things he's interested in.)

ASSIGNMENTS WEEK 2

1. QUALITY ATTRIBUTES (20 PT)

- Have a look at chapters 16 and 17 from MS Application Architecture Guide. What is the **difference** between quality attributes and crosscutting concerns?
- What is the **difference** between a Perspective and a Viewpoint as defined in *Software Systems Architecture*?
- Describe two quality attributes that **typically conflict**, i.e. more of one attribute means less of the other. In what way can you make tradeoffs between them? (Don't choose the examples mentioned at the lectures.)

2. ARCHITECTURAL PATTERNS (20 PT)

- Describe in your own words what the **difference** is between design patterns as taught in the previous period (the Gang-of-Four patterns) and architectural patterns.
- Design patterns and architectural patterns are often based on the same ideas. Which design patterns are **most similar** to the architectural patterns *Pipes and Filters* and *Indirection*? (Pipes and filters is described on page 140-142 of *Software Systems Architecture*).
- Give an example of the **application** of the Pipes and Filters-pattern.
- Lookup an architectural pattern of your choice and **describe it in your own words**. You are not allowed to copy/paste any text from existing sources. Use the same format as the example description of the Shared Repository-pattern given on Blackboard. Also, have a look at the article *A Pattern Language for Pattern Writing* (especially patterns B.1 and B.2) for an explanation of the pattern elements in the template.

3. CREATING A SAD FOR A NEW SYSTEM (60 PT)

This case study is about the intelligent monitoring of mentally handicapped people. It is based on a real-life project in the research group New Business & ICT within our school. Blackboard provides you with the initial project proposal (NOVO LVG) which was used to get a grant from the province of Groningen. A version of the Description of Work is also available. The system that was built was meant to be used for the pilot phase. If the test results turn out to be promising, a company would extend it and market it.

Study the available documentation and design a software architecture for such the system. Write it down in a SAD using the Viewpoints/Perspectives from Rozanski & Woods.

Some pointers:

- What are the stakeholders and their concerns?
- What are the relevant viewpoints and perspectives?
- What are the *architecturally significant* functional and non-functional requirements?
- What are the major tradeoffs you have to make when trying to satisfy competing requirements for this system? Document your critical Design Decisions with alternatives and rationale.
- Consider technology choices for several areas. Research which options you have and again present a substantiated choice in the form of a design decision.
- Which 'perspectives' are the most relevant for the system?
- Which tactics do you use from the perspectives and why?

- How can you be confident that you will meet the significant requirements? Is it a matter of convincing argumentation? Or do you need to test some stuff? ¹
- Give a suggestion for the sequence in which the development of the system can be done. (This would normally be part of the Software Development Plan, but we won't ask you to write one now.)

ASSIGNMENTS WEEK 3

1. PEER-TO-PEER (50 PT)

The application responsible for most of the internet traffic is probably peer-to-peer (P2P) file sharing. The most popular P2P protocol is the bittorrent protocol. More on the bittorrent protocol can be found on [the-bittorrent-protocol](#) and [wiki](#). Also have a look again at the XChange assignment.

- a) Can you mention some other popular P2P file sharing protocols ?
- b) What are the stakeholders in a peer-to-peer network like XChange? What are their concerns?
- c) What are the advantages and disadvantages of P2P filesharing compared to client/server file sharing? Discuss at least the 5 major perspectives (from the SSA book).
- d) Why do P2P networks have excellent scalability? Give numerical examples and graphs to support your answer.
- e) Explain in what way the integrity of (shared) files is preserved.
- f) Explain what is meant by torrent index and tracker. And why is it difficult to forbid torrent indexes and trackers ?
- g) Evaluating availability, what are in your opinion the weak points in the architecture of the protocol ?
- h) Can you think of a design pattern that is comparable to the role of a tracker in the bittorrent protocol ?
- i) Can you imagine how the architecture of bittorrent could be applied to a complex application or system ? If so, give an example.
- j) Explain in your own words the working of a bittorrent peer-to-peer system by drawing deployment and sequence diagrams. Supplement the drawings with text. Go into depth about which data is on which node, hashes, etc.

2. ENTERPRISE ARCHITECTURE (50 PT)

There questions concern the article *Enterprise Architecture for Systems Engineers* by IBM.

- a) Explain in your own words what "stovepipe systems" are (p.1).
- b) Explain in your own words what "business/IT alignment" is (p.1).
- c) Explain in your own words what "governance" is (p.1).
- d) Explain in your own words the relationship between programs and projects (p.3).

From page 5 onwards the article gets a little abstract because the authors don't give an example. We will use the one below.

¹ It's not forbidden to test some technology or intended solution BUT only if time allows it. If time doesn't allow it then say how you would test it if you had more time. Finishing your SAD has higher importance.

The RDW (“Dienst Wegverkeer”, www.rdw.nl) is a Dutch governmental organization which registers cars, drivers licenses, etc. They currently have a large legacy base on mainframe/Cobol (both databases and applications). One of their programs, which is called Platform Independence, is to migrate from the mainframe to a Microsoft platform.

- e) Give three reasons why the RDW would want to migrate away from the mainframe.
- f) Give two reasons why the RDW would *not* want to migrate away from the mainframe.
- g) Which one of the strategic goals on <http://www.rdw.nl/overrdw/Paginas/Strategische-doelstellingen.aspx> is the main driver for this program?
- h) Think of three possible projects that could be part of the program Platform Independence.