

TDA 594/DIT 593 - Individual Assignment

Due Date: Friday, January 13, 23:59

Submission: Via Canvas (one PDF document per student)

Overview

Now that you are completing the group project and have seen how SPL development affects all stages of the development lifecycle, we would like you to reflect on what you have learned in an individual assignment. This assignment tasks you with reflecting on both the theoretical and practical course material.

Task 1: Comparative Study of Multiple Software Product Lines

In the first group assignment, you created a case study examining the development of a Software Product Line. In this task, you will write a “comparative study” where you revisit the case study from Assignment 1 and read an additional case study. **Please choose this additional case study from the list available as part of Assignment 1.**

Write a short report summarizing, comparing, and contrasting the two case studies. Your report should comment on as many of the following aspects as possible:

- **Context:** For each study, what kind of organization adopted/applied SPL engineering? Contrast the types of organizations (How did they differ? What did they have in common?)
- **Motivation:** For each study, what motivated the transition to, or adoption of, a product line? How are the motivations similar or different?
- **System Type:** For each study, what kind of system did they apply SPL engineering to? Specify the application domain (e.g., embedded system, database management...) and, if available, the size and scope of the project (e.g., number of variants, number of source code files). Contrast these factors across the two studies.
- **Approach:** For each study, how did they develop or transition to a SPL? What development practices were employed? What development processes or activities (e.g., requirements, design, implementation, or testing activities) were affected, and how? What were the commonalities and differences in the approaches across the two studies.
- **Challenges:** For each study, what were the key technical or process challenges encountered when implementing the SPL? Which challenges were shared or differed across the two studies?
- **Results:** For each study, what are the important results with regard to business goals, software architecture (e.g., its ability to meet goals or requirements, changes in maintainability and changeability), development process, and development organization? Compare and contrast the results. Reflect on why results may have differed (positively or negatively).

You should attempt to find information regarding each of the above aspects. State explicitly if information is not available for an aspect. In such situations, you may speculate on aspects of the development of the system, but must make clear what you are assuming or speculating.

You **must** reflect on the choices made by the engineers building these systems and **provide your own commentary and opinions** on those choices. In your view - given the knowledge you have gained in this class - was the approach the developers adopted a good one, given their development context? If you believe they made the best decision, explain your rationale. Otherwise, do you have advice on approaches they could have considered instead? Do you see potential weaknesses in the decisions made? Are there alternatives you feel should have been considered? Discuss the work performed by these companies in the context of your own experiences or in the context of other systems that you have read about.

Task 2: Design of a Software Product Line

Consider the following example. **Apo-Games** is a set of five different Android games for which the source code is publicly available. These five games have been published within two years and contain between 2,000 and 6,500 source lines of code. Each game uses a customized version of a third-party game engine.

| Name | Year | Lines of Code | Game Type |
|------------|------|---------------|---------------------|
| ApoClock | 2012 | 3615 | Arcade/Puzzle |
| ApoDice | 2012 | 2523 | Level-based Puzzles |
| ApoSnake | 2012 | 2965 | Snake |
| ApoMono | 2013 | 6487 | Level-based Puzzles |
| MyTreasure | 2013 | 5,360 | Level-based Puzzles |

- ApoClock provides two different game modes, arcade and puzzle. In both modes, the player has to hit clocks with a ball before any of the clocks runs out of time. While the arcade mode is endless and accumulates a score, the goal in the puzzle mode is to clear a designed level.
- ApoDice is a puzzle game that shows dice to the player, with each die having a number indicating how often it can move. The player has to move each die to a black square on the board within the given number of moves. A total of 30 levels are predefined, but users can add their own.
- ApoMono requires the player to move an avatar from a starting position to a goal. Throughout the avatar's path, there are different obstacles, for example, missing tiles or walls. The player can move some stones on the field to overcome such obstacles.
- ApoSnake is similar to the original Snake game, where the player controls a snake to collect candy, increasing its size. However, in ApoSnake, the amount of candies in a

level is fixed and they have different colors, which color the snake in the same way. The snake can then move through any wall that has the same color as itself.

- MyTreasure is a puzzle game within a two dimensional maze. The player has to collect a golden coin in this maze, for which they have to rotate the maze, allowing the avatar to move according to gravity. In addition, yellow blocks in the maze also move according to the current rotation.

All of these games also have an editor that allows the user to create their own custom levels for that specific game. Custom levels are stored together with all levels that other users created, and all levels can be retrieved and played by using a button in the menu.

You can read more about Apo-Games at <https://dl.acm.org/doi/abs/10.1145/3233027.3236403> or <http://apo-games.de/> (you can auto-translate from German in Chrome or using Google Translate).

The developers of these games want to transition to a Software Product Line where they can produce new games or variants of existing games with different configurations from a shared set of common assets. Your task is to perform domain analysis and explain how this transition should take place.

1. Perform domain analysis and produce a feature diagram for this new SPL.
 - In addition to the feature diagram, briefly explain the features you have identified.
2. Explain how you think that variability should be implemented in this SPL for the features that you identified.
 - Choose among preprocessors, design patterns, a framework, components, feature-oriented programming, and aspect-oriented programming. You may choose to blend implementation mechanisms, as long as you can make a reasonable argument for your choices.
 - Explain why you have chosen these particular implementation mechanisms for the affected features, and explain how the specific features you have identified could be implemented using the chosen mechanisms. Prepare a minimal skeleton code example to illustrate how you would apply this technique.
 - For example, if you choose design patterns, explain which design patterns you would apply for a particular subset of features and why those patterns are the correct ones for those features, and illustrate how those design pattern could be applied using a minimal code outline.
 - Do not simply quote lecture materials, but relate those materials to the specific features you have identified.
 - We do not expect executable code, but an illustration of how you would apply the selected mechanisms.
 - In addition to the information provided above, you can view the source code at https://bitbucket.org/Jacob_Krueger/apogamessrc/src/master/. You can also attempt to install and play a subset of the games on an Android device from <https://play.google.com/store/apps/developer?id=Apo-Games&hl=en&gl=US&pli=>

1. You do NOT have to build, run, or deeply analyze the existing source code. It is available to help inform your opinions.
 - You are welcome to make your own assumptions about how each game works - or how the theoretical SPL would work - as long as you are clear about what you are assuming and what has been extracted from the source code or other documentation.
3. Choose two interesting variants (valid configurations) and contrast them.
 - State each configuration.
 - Explain your view of how each variant would function and how it would differ from the other selected variant. Describe the functional differences in detail. For example, what particular use cases or market segments would this variant serve that the other selected variant could not? What capabilities would this variant have that the other would not?

Additional Notes

This is an **individual** assignment. All work must be your own. If you quote another source, you must cite it properly. You must not work with other students in the course. We will perform plagiarism checks, and any instances of plagiarism or working together will result in (at minimum) a failing grade on this assignment. If you are not sure if something will break these rules, ask us!

There is not a minimum page length for this document. Quality is more important than quantity. It is important that you cover as many of the criteria as possible in detail, and provide your own reflections on the development of the chosen system.

Submit your comparative study in **PDF form** via the submission link on Canvas.

Grading Guidelines

These guidelines are intended to give some guidance, but are not exhaustive. The examiner will assign a grade based on the correctness and quality of your work.

| Grade | Guidelines |
|-------|--|
| 5 | <ul style="list-style-type: none"> ● Task 1: <ul style="list-style-type: none"> ○ Covers the full set of stated aspects above for both studies, as well as additional aspects regarding the development of the system that you have found interesting. ○ Each aspect is covered in detail and contrasted between the two studies. The comparison includes your original commentary and creative reflection. ○ If some aspects cannot be covered, informed speculation and discussion is still provided. |

| | |
|---|---|
| | <ul style="list-style-type: none"> ● Task 2: <ul style="list-style-type: none"> ○ Thorough, well-documented, and correctly-formatted feature model. Detailed explanations of the identified features. ○ Clear and detailed explanation of chosen implementation mechanisms. Explanation relates to the specific features identified, with specific (skeleton) code examples and other illustrations of how implementation could be accomplished. ○ Chosen variants clearly explained and contrasted in detail, with clear use cases and feature differences. |
| 4 | <ul style="list-style-type: none"> ● Task 1: <ul style="list-style-type: none"> ○ All aspects covered. ○ Most aspects are covered in detail and contrasted. Comparison includes your original commentary and creative reflection. ○ If some aspects cannot be covered, informed speculation and discussion is still provided. ● Task 2: <ul style="list-style-type: none"> ○ Thorough, reasonably-documented, and correctly-formatted feature model. Understandable explanations of the identified features. Small mistakes may be present, or slightly fewer features may be present than a “5”. ○ Detailed explanation of chosen implementation mechanisms. Explanation relates to the specific features identified, with specific (skeleton) code examples. ○ Chosen variants explained and contrasted in detail, with an attempt to explain use cases and features. |
| 3 | <ul style="list-style-type: none"> ● Task 1: <ul style="list-style-type: none"> ○ Most aspects covered. ○ Some aspects covered in detail and contrasted, with original commentary and creative reflection. ○ If some aspects cannot be covered, some speculation and discussion is still provided. ● Task 2: <ul style="list-style-type: none"> ○ Documented and correctly-formatted feature model. Some explanation of the identified features. Small mistakes may be present, or slightly fewer features may be present than a “4”. ○ Some explanation of chosen implementation mechanisms. Explanation relates to the specific features identified, with skeleton code examples. ○ Chosen variants explained and contrasted, with an attempt to explain use cases and features. |

U

- Task 1:
 - Several aspects missing from the above list.
 - Adequate contrast between studies not provided.
 - Included aspects not covered in sufficient detail.
 - No original commentary or reflection.
- Task 2:
 - Undetailed feature model of many mistakes in the feature model. Lack of explanation of identified features.
 - Explanation of chosen implementation mechanisms lacking. Explanation relates to the specific features identified, with specific (skeleton) code examples.
 - Chosen variants explained and contrasted in detail, with an attempt to explain use cases and features.