Thant Nyi Nyi Thaw SAD Assignment

by Thant Nyi

Submission date: 21-Jul-2023 03:17PM (UTC+0800)
Submission ID: 2134451955
File name: SAD.docx (466.79K)
Word count: 968
Character count: 5891

COS-107 – Information Systems Analysis and Design

Assignment

HDIT

Thant Nyi Nyi Thaw

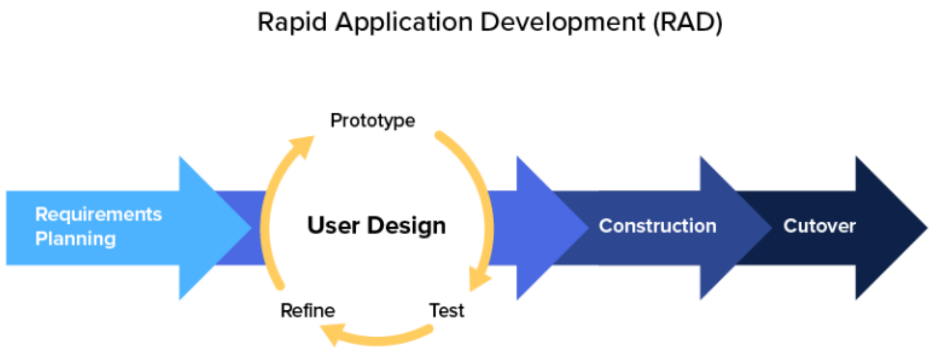
7/20/2023

Table of Contents

Task 1

What is RAD?

Rapid Application Development, or RAD, is a flexible approach to software development that places less of a focus on detailed design and more on experimentation and quick feedback. In general, the RAD approach places more emphasis on prototyping and development than it does on planning. Rapid application development enables developers to swiftly iterate and update software



without having to start from scratch. This ensures that the final product is more quality-focused and meets the needs of the end consumers.The ability to modify the design, add functionality, and iterate as frequently as feasible without having to start over is one of the main benefits of rapid application development (Anon., 2023).

Types of RAD:

1. **Incremental RAD:** Incremental RAD breaks down the development process into smaller, manageable increments. Each increment delivers a functional part of the system. It involves rapid prototyping and iterative development, allowing stakeholders to provide early feedback on each increment. This approach is suitable when the project scope is well-defined, and there is a need for regular releases of working software.
2. **Evolutionary RAD:** Evolutionary RAD focuses on building a basic, functional system as quickly as possible. The system is then refined and expanded through successive iterations. This approach is ideal for projects with evolving requirements or when the time-to-market is crucial. Stakeholders can prioritize features to be added in subsequent iterations, ensuring that the most critical functionalities are delivered first.

Recommendation and Justification:

As a systems analyst, I would recommend considering the RAD approach in certain scenarios. It is essential to evaluate the project's characteristics and constraints before deciding on the development methodology. Here are some justifications for recommending or not recommending RAD:

Recommendation:

1. **Small to Medium-Scale Projects:** RAD is well-suited for small to medium-sized projects with well-defined and achievable goals. Its rapid delivery and flexibility align well with such projects.
2. **Dynamic Requirements:** If the project involves dynamic or changing requirements, RAD's iterative approach can accommodate these changes effectively.
3. **User Involvement:** When user involvement is critical for project success, RAD's emphasis on continuous user feedback can ensure the final product meets user expectations.

Caution:

1. **Large-Scale Projects:** For large-scale, complex projects with extensive requirements, RAD might not be the best fit. Its focus on speed could compromise system stability and scalability.
2. **Regulated Environments:** Projects operating in heavily regulated industries may require more stringent documentation and compliance, which may not align well with RAD's fast-paced development.
3. **Resource Constraints:** RAD requires skilled and dedicated team members. If resource availability or budget constraints are significant, RAD might not be the most feasible choice.

Strengths of RAD:

1. **Speed of Development:** RAD emphasizes rapid prototyping and iterative development, allowing software to be delivered quickly. This is particularly beneficial for projects with tight deadlines or changing requirements.

2. **Client Involvement:** RAD encourages active client participation during the development process. This involvement ensures that the final product meets the client's needs and expectations effectively.
3. **Flexibility:** RAD allows for incremental changes and additions throughout the development process, making it easier to accommodate changing requirements and business needs.
4. **Cost-Effectiveness:** By promoting early feedback and continuous testing, RAD reduces the risk of costly errors and rework during later stages of development.
5. **Improved Quality:** Rapid prototyping and early user feedback help identify issues and improvements early in the development process, resulting in a higher-quality end product.
6. **Improved Communication:** RAD's emphasis on collaboration and teamwork fosters better communication between developers, stakeholders, and end-users.

Weaknesses of RAD:

1. **Complexity Management:** Rapid development can lead to increased complexity in the codebase, making maintenance and future updates more challenging.
2. **Resource Intensive:** RAD requires skilled and experienced team members, which can be more expensive to assemble than traditional development teams.
3. **Dependency on User Availability:** Continuous user involvement is crucial in RAD. If users are not available or engaged throughout the process, it can lead to delays and misalignment with requirements.
4. **Limited Scope:** RAD may not be suitable for large-scale projects with complex requirements or extensive regulations.
5. **Lack of Documentation:** Due to the focus on speed, documentation might be neglected, making it harder for future developers to understand and maintain the system.

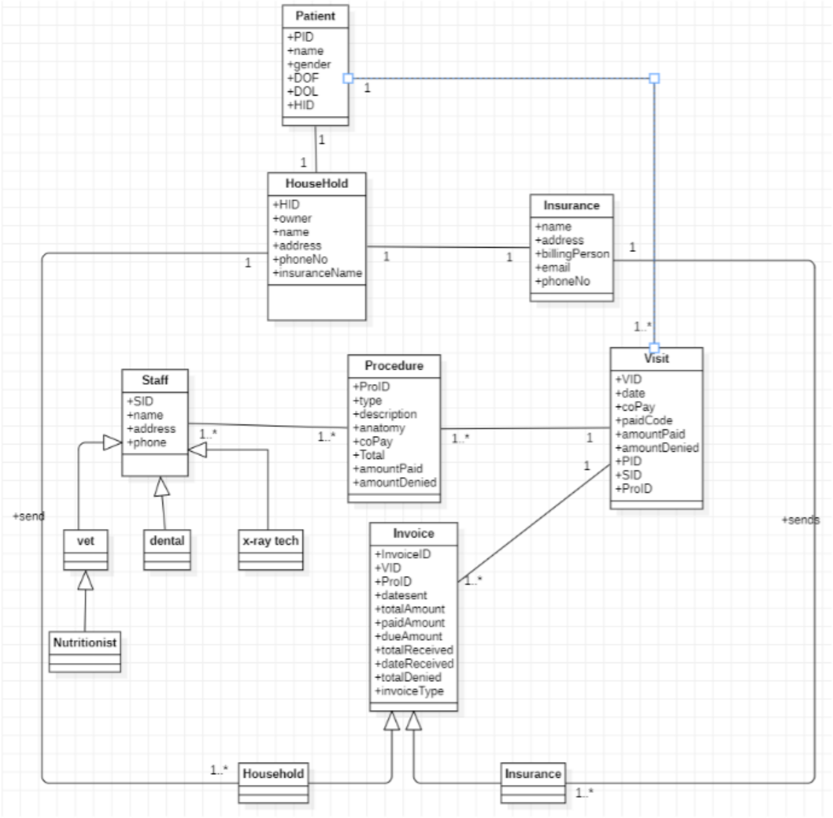
Relevancy of RAD in Today's Information Systems Development:

RAD remains relevant in today's information systems development due to several factors:

1. **Agile Development:** RAD shares similarities with Agile methodologies, which have gained significant popularity in recent years. Both prioritize customer collaboration, iterative development, and incremental delivery.
2. **User-Centric Approach:** With the growing importance of user experience (UX) in software development, RAD's emphasis on early user feedback aligns well with modern development practices.
3. **Rapid Technology Advancements:** The availability of advanced software tools and frameworks further supports RAD's rapid prototyping and development capabilities.
4. **Competitive Markets:** In fast-paced industries, RAD allows organizations to quickly deliver working software, gaining a competitive advantage.

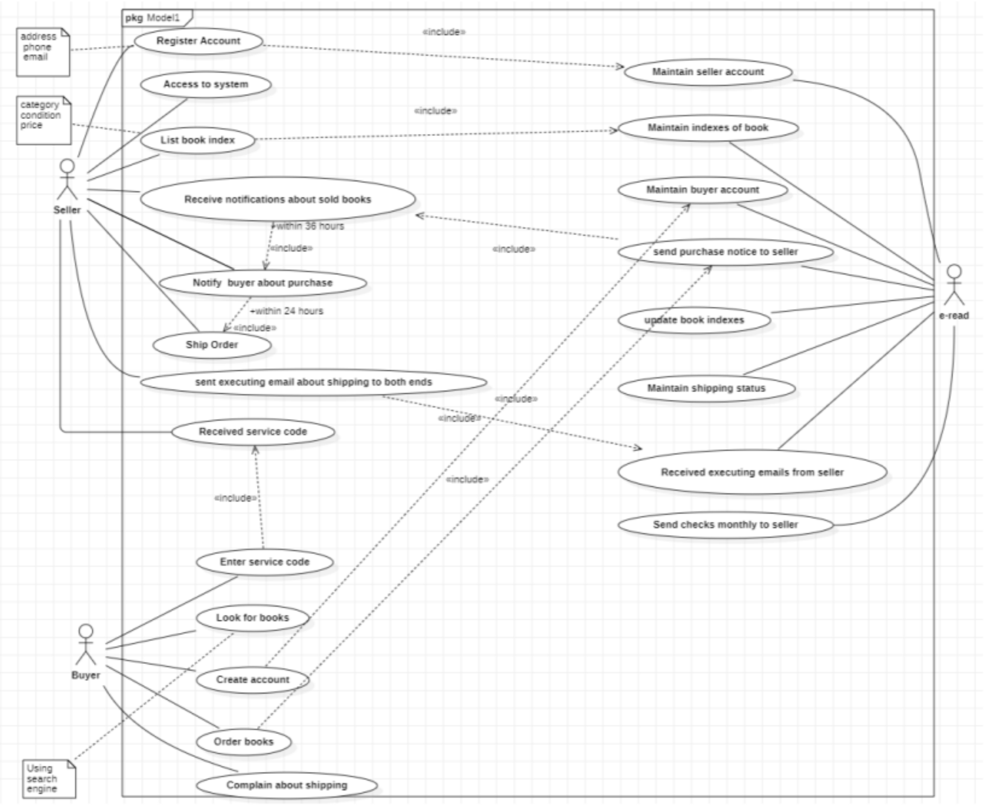
In conclusion, RAD remains a relevant and valuable approach for specific projects, especially those with smaller scopes, dynamic requirements, and a need for rapid delivery. As a systems analyst, it is crucial to carefully assess the project's characteristics and consult with stakeholders before recommending RAD or any other development methodology. By understanding the strengths and weaknesses of each approach, analysts can make informed decisions to ensure successful project outcomes.

Task 2



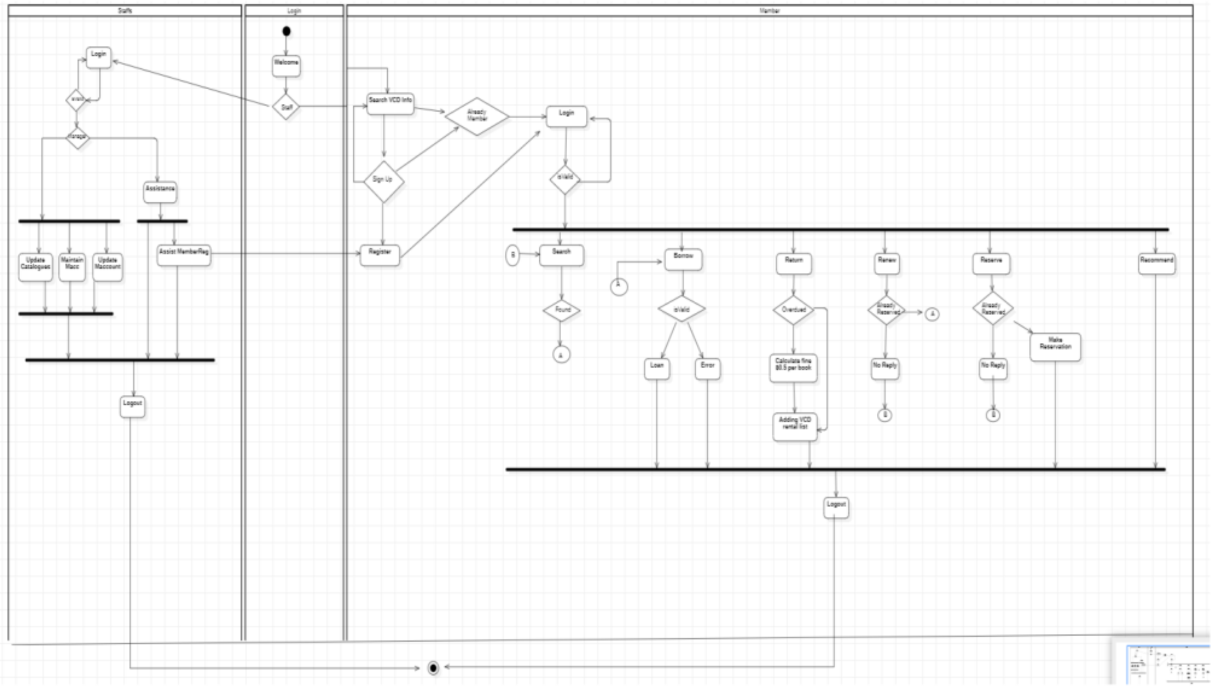
This Class diagram is drawn by Star UML

Task 3



This Use Case diagram is drawn by Star UML

Task 4



This Activity diagram is drawn by Star UML.

Bibliography

Anon., 2023. *kissflow.* [Online]
Available at: https://kissflow.com/application-

development/rad/rapid-application-development/
[Accessed 20 7 2023].

Thant Nyi Nyi Thaw SAD Assignment

ORIGINALITY REPORT

20%	2%	1%	18%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Greenwich Student Paper	10%
2	Submitted to TMC Education Group Student Paper	7%
3	Submitted to Harare Institute of Technology Student Paper	2%
4	www.coursehero.com Internet Source	1%
5	Evan W. Duggan. "chapter 12 Tranquilizing the Werewolf that Attack Information Systems Quality", IGI Global, 2006 Publication	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		