

New Software Development Paradigms Exam Memo

Open book
100 marks
Three hours

Question 1

Using an example of your choice, explain iterative software development.

[10]

Answer:

2 marks per block [example and description] and then 2 if they understand the different builds (full MVP)

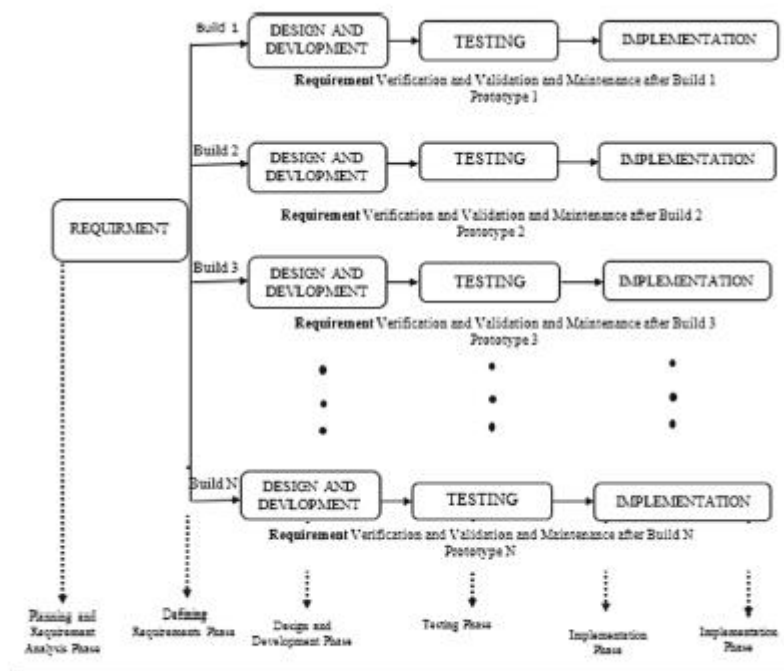


Fig. 3. Importance of requirement in the Iterative SDLC model [6]

Question 2

You are the Scrum master for a team that are building a new music sharing application. Give examples of what will the different Scrum Events and Artifacts will consist of.

[20]

Answer:

Scrum Events	7
The Sprint	7
Sprint Planning	8
Daily Scrum	9
Sprint Review	9
Sprint Retrospective	10
Scrum Artifacts	10
Product Backlog	10
Commitment: Product Goal	11
Sprint Backlog	11
Commitment: Sprint Goal	11
Increment	11
Commitment: Definition of Done	12

Question 3

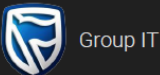
Alan Gray Financial Services had a system outage for four days that costed them a lot of good will and new business. They have hired you to advise them on how to make their system more robust to failure. Use **practical examples** to tell the management what you will implement to make the systems more reliable










[18]

Answer

2 marks per block (description and example)

RESILIENCE REQUIRES SYSTEMS THINKING



 <p>Infrastructure as Code Without infrastructure as code configuration drift <i>cannot</i> be managed.</p>	 <p>Pipelines Automated builds, automated testing, consistency across environments. Compliance & security built-in.</p>	 <p>HA/DR Base-level requirements for availability. Necessary but insufficient for continuous availability.</p>
 <p>Observability Inferring internal state from knowledge of external outputs. See the accident coming.</p>	 <p>Incident Management Detect, respond, alert, communicate, manage.</p>	 <p>System Teams Providing the tools and practices for building and maintaining resilient systems. Non-functional Kung fu masters.</p>
 <p>Blast Radius Loose coupling, isolate failure, graceful failure.</p>	 <p>Communication Keeping the Bank, Customers and Regulators informed at the time of the incident and during service recovery.</p>	 <p>Retrospectives Blameless post mortems, learning, sharing, educating.</p>

Question 4

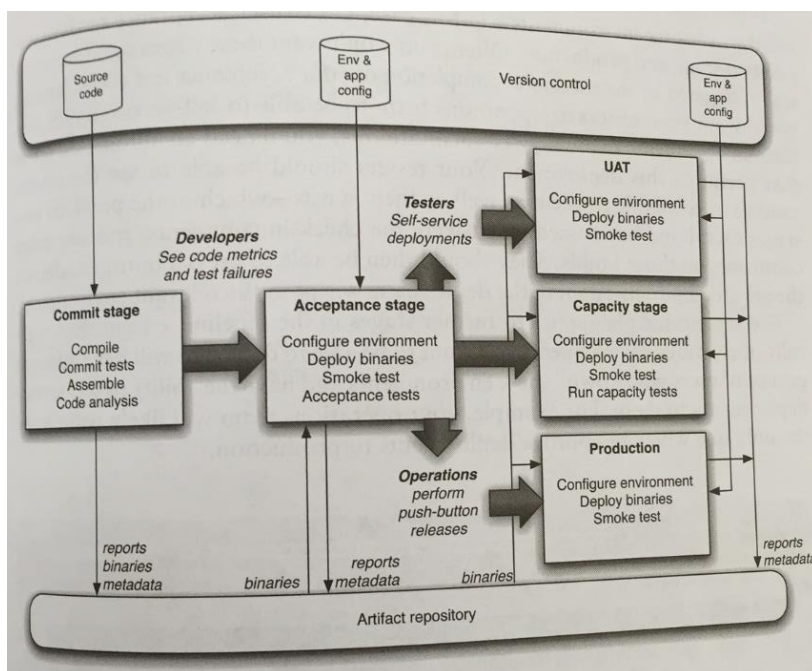
A deployment pipeline is an automated manifestation of the process for getting software from version control into the hands of the users.

- a) Using an example of a software development project of your choice describe how you will implement a DevOps pipeline. Focus on the tools and processes. Use the work of Humble (Humble & Farley, 2011). to guide your thinking

[14]

Answer 4

Give one mark for each block and another for an example of it from the following diagram up to a maximum of 14 marks.



Question 5

You have been asked to interview a Design Thinking expert for your company. What characteristics would you look for and what questions would you ask to see if they have those characteristics.

[10]

Answer 5:

A Design Thinker's Personality Profile

Contrary to popular opinion, you don't need weird shoes or a black turtleneck to be a design thinker. Nor are design thinkers necessarily created only by design schools, even though most professionals have had some kind of design training. My experience is that many people outside professional design have a natural aptitude for design thinking, which the right development and experiences can unlock. Here, as a starting point, are some of the characteristics to look for in design thinkers:

Empathy. They can imagine the world from multiple perspectives – those of colleagues, clients, end users, and customers (current and prospective). By taking a "people first" approach, design thinkers can imagine solutions that are inherently desirable and meet explicit or latent needs. Great design thinkers observe the world in minute detail. They notice things that others do not and use their insights to inspire innovation.

Integrative thinking. They not only rely on analytical processes (those that produce either/or choices) but also exhibit the ability to see all of the salient – and sometimes contradictory – aspects of a confounding problem and create novel solutions that go beyond and dramatically improve on existing alternatives. (See Roger Martin's *The Opposable Mind: How Successful Leaders Win Through Integrative Thinking*.)

Optimism. They assume that no matter how challenging the constraints of a given problem, at least one potential solution is better than the existing alternatives.

Experimentalism. Significant innovations don't come from incremental tweaks. Design thinkers pose questions and explore constraints in creative ways that proceed in entirely new directions.

Collaboration. The increasing complexity of products, services, and experiences has replaced the myth of the lone creative genius with the reality of the enthusiastic interdisciplinary collaborator. The best design thinkers don't simply work alongside other disciplines; many of them have significant experience in more than one. At IDEO we employ people who are engineers *and* marketers, anthropologists *and* industrial designers, architects *and* psychologists.

Question 6

You have been appointed as an advisor to the board of Nedbank. They have contracted you to give recommendations on the use of non-financial data in their organisation and to augment it with practical examples.

[20]

Answer:

<p>RECOMMENDATION 1: DISCLOSURE AND INFORMED CONSENT.</p> <p>FSPs should be clear about their use of customer data, attain customer agreement to their customer data policies and, where appropriate, seek consent for specific uses, e.g.</p> <ul style="list-style-type: none"> • Informed consent: FSPs need to provide clear and accessible information about how customer data will be used (e.g. terms and conditions). • Transparency: Customers should be able to view or know the data that are collected about them, how they are used and whether they are shared with a third party. • Ability to revoke consent: Customers should be able to request that data about them no longer be used by an FSP (e.g. the right to be forgotten). • Legitimate use: FSPs may not need to seek consent when using data for legitimate interests (e.g. those required by law). 	
<p>RECOMMENDATION 2: SECURITY</p> <p>FSPs should be held responsible and accountable for data security, e.g.</p> <ul style="list-style-type: none"> • Liability: A clear liability framework should be in place that ensures the responsible party is held accountable for data security and harms caused by breaches of its respective data security duties of care. • Traceability: FSPs need to be able to identify where data were improperly used or accessed in the event of a security breach. 	<p>RECOMMENDATION 3: CONTROL</p> <p>FSPs should disclose to customers which of their data points they are using and enable customers to intervene and limit use where applicable, e.g.</p> <ul style="list-style-type: none"> • Intervention: Customers should be able to intervene to gain information or limit the use of data they control, and FSPs should respond appropriately. • Limited use: Where reasonable, a maximum time period that data can be retained by FSPs should exist, as well as limits on certain sensitive data types or uses.
<p>RECOMMENDATION 4: PORTABILITY.</p> <p>FSPs should, where appropriate, allow customers to access, download, transfer and/or permit third parties to manage data about them, e.g.</p> <ul style="list-style-type: none"> • Accessibility: FSPs should allow customers to download data about them in a machine-readable format or through standardized APIs, depending on the FSP stage of development. • Third-party permissions: Customers should permit third parties to download their data. 	
<p>RECOMMENDATION 5: PRIVACY AND DATA MISUSE.</p> <p>FSPs should be held responsible and accountable for violation of customers' data privacy, e.g.</p> <ul style="list-style-type: none"> • Liability: A clear liability framework should be in place that ensures the responsible party is held accountable for data misuse and harms caused by breaches of its respective data duties of care. • Traceability: FSPs need to be able to identify where data were improperly used. 	
<p>RECOMMENDATION 6: ALGORITHMS AND ANALYTICS</p> <p>FSPs should be able to comprehensively test, validate and explain their use of data analytics or algorithms and models to customers, e.g.</p> <ul style="list-style-type: none"> • Justification: Customers should have the right to request why a decision was made (e.g. why the model methodology is appropriate, why the output is justified). • Challenge: Customers should have the right to correct incorrect or incomplete data about them held by an FSP. 	

Question 7

Contrast and compare the Rational Design Paradigm Meta-narrative with that of the Empirical Design Paradigm Metanarrative

[8]

Answer 7:

A maximum of 8 marks based on the following:

Box 1 Illustrative Metanarratives of the Two Paradigms

The Rational Design Paradigm Metanarrative	The Empirical Design Paradigm Metanarrative
<p>The problem is known and the goal of the system is clear. Analysts elicit comprehensive, unambiguous requirements which are agreed by the client. Designers search a conceptual solution space for design candidates that satisfy the requirements. They use logic and reason to deduce an appropriate architecture or user interface. Design decisions are concentrated in this phase of the project. Developers select an appropriate software development method and use it to build the system. Although perfect rationality is impossible, developers strive to be as structured, methodical and rational as they can. They plan development as a series of activities or phases with milestones and execute this plan. Unexpected events trigger re-planning. Teams understand and evaluate their progress in terms of the plan. The project is successful if it delivers the agreed scope within the allotted time and budget, at a reasonable quality. Researchers understand this process in terms of lifecycle models and software development method use.</p>	<p>There is no "the problem." There is a situation that different stakeholders (with different goals) perceive as problematic in different ways. Analysts work with stakeholders to collaboratively construct ideas and preferences for a possible system. It is not clear which of these ideas and preferences are requirements because no one knows for sure whether each feature is critical, optional, or counterproductive. Understanding of the problematic situation and possible design candidates coevolve: ideas about solutions trigger problem reframing, which triggers new solution ideas, and so on. Designers rely on creativity and intuition. Design pervades the project, with key properties emerging during from users, designers and developers during interviews, analysis, programming, refactoring, etc. Each project presents unique sequences of events, which do not necessarily resemble known methods or process models, and unexpected events are common. Plans and software development methods and consequently weak resources for informing behaviour, so people improvise. Project success is complicated and controversial but making the problematic situation better in the eyes of its stakeholders generally outweighs technical performance and meeting contract terms. Researchers understand this process in terms of teleological and dialectical process theories, as well as professional behavior.</p>