Module 1 – Topic 1.2

Foundations of Managing Software Quality and Process

Topic Outline

- Project Success and Quality Factors
- Quality in the Lifecycle
  - Right product
  - Done right
  - Managed right
- Monitoring, Metrics and Feedback
- Why Agile?
- Career Opportunities

How Would You Define Software Development Success?

Choose the factor or factors that you think are most representative of a successful project

A. On schedule
B. On budget
C. To specification

Reminder: Reading 1.1 Project Success Survey by Scott Ambler and Associates in 2013

There is no common definition of software development success!

Which is a Successful Car?

Discuss

Why?

Which is a Successful Course?

- COMP1100 / COMP1130 - Programming as Problem Solving - Sem 1 2017
- COMP1710 / COMP6780 - Web Development and Design - Sem 1 2017
- COMP2100 / COMP6442 - Software Design Methodologies - Sem 1 2017
- COMP2140 / COMP6700 - Java Programming - Sem 1 2017
- COMP2300 / COMP6300 - Computer Organisation and Program Execution - Sem 1 2017
- COMP2410 / COMP6340 - Networked Information Systems - Sem 1 2017
- COMP2550 / COMP4450 - Advanced Computing R&D Methods - Sem 1 2017
- COMP3100 / COMP3500 / COMP3550 / COMP4500 / COMP8715 - Software Engineering Group Project - Sem 1 2017
- COMP3120 / COMP8110 - Managing Software Development - Sem 1 2017
- COMP3530 / COMP6353 - Systems Engineering for Software Engineers - Sem 1 2017
- COMP3560 - Advanced Computing R&D Industry Experience - Sem 1 2017
- COMP3620 / COMP6320 - Artificial Intelligence - Sem 1 2017
- COMP3710 - Topics in Computer Science - Sem 1 2017
- COMP4130 - Managing Software Quality and Process - Sem 1 2017

Why?

Discuss

This one ☺?

Which is a Successful OS?

Why?

Discuss

MacOS™

Microsoft® Windows®

What is Quality?

- Statement 1
  - Quality is a characteristic of thought and statement that is recognised by a **non-thinking** process.
    Because definitions are a product of right, formal thinking, *quality cannot be define*.

  - However, even though quality cannot be defined, *you know what quality is!*

What is Quality?

- Statement 2
  - Quality is *what you like*
- Statement 3
  - Quality is *subjective* – existing only in the observer?
    - Just another way of saying it's whatever you like.
- Statement 4
  - Quality is *objective* – but is it measurable and provable?
    - What scientific instruments can be used to measure quality?

What is Quality?

- Pirsig's dilemma:
  - If quality exists in the object, then you must explain just why scientific instruments are unable to detect it …
  - On the other hand, if quality is subjective, existing only in the observer, then this quality is just a fancy name for whatever you like …
  - Quality is not objective. It doesn't reside in the material world
  - Quality is not subjective. It doesn't reside merely in the mind

What is Software Quality?

- Statement 1
  - Quality is *value to some person*
- Statement 2
  - Quality is *conformance to requirements*
- Statement 3
  - Quality is the *absence of error*

What is Software Quality?

- And there is more! Quality is:
  - Having lots of short-cut features in the interface
  - Easy to use, intuitive interface
  - A context sensitive help system
  - High performance
  - An understandable design
  - Small amounts of documentation
  - Low maintenance cost
  - High productivity
  - To budget and schedule
  - ……

What is Software Quality?

- IEEE Standard Glossary of Software Engineering Terminology:

  The degree to which a system, component, or process meets
  - Specified requirements, and
  - Customer or user needs or expectations

FURPS+ for Defining Quality

- Functionality
- Usability
- Reliability
- Performance
- Supportability

Plus:

- Design constraints
- Implementation req'ts
- Interface req'ts
- Physical req'ts

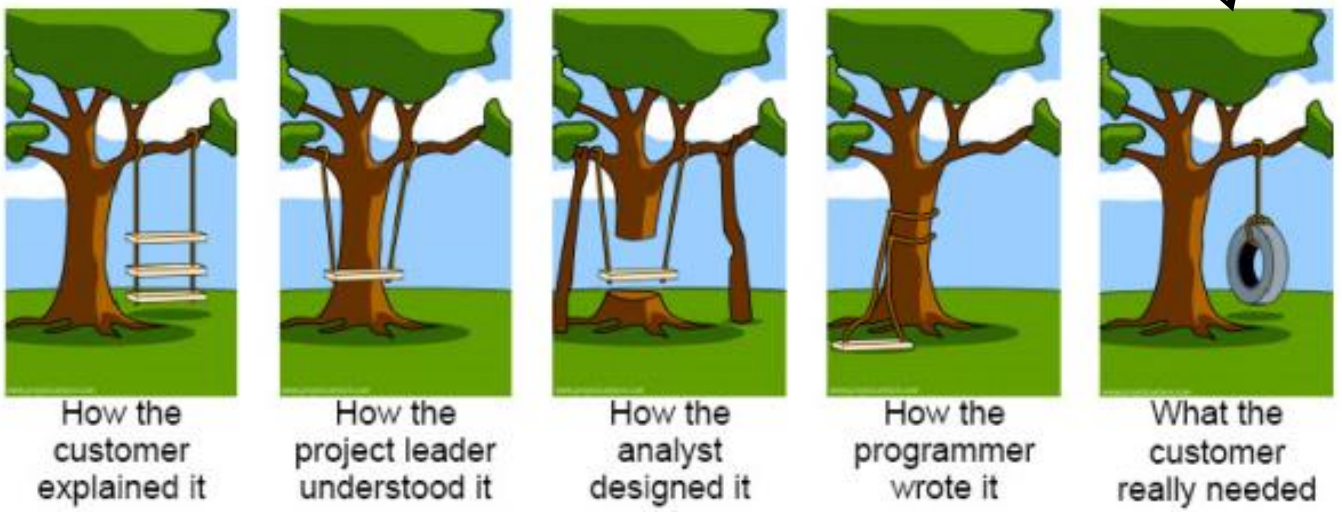
Devised by Robert Grady of HP

- Product quality and stakeholder value
- FURP for user
- S for developers, testing, maintainers

Functionality

- What the customer needs
  - Feature set
  - Security – safety, exploitability

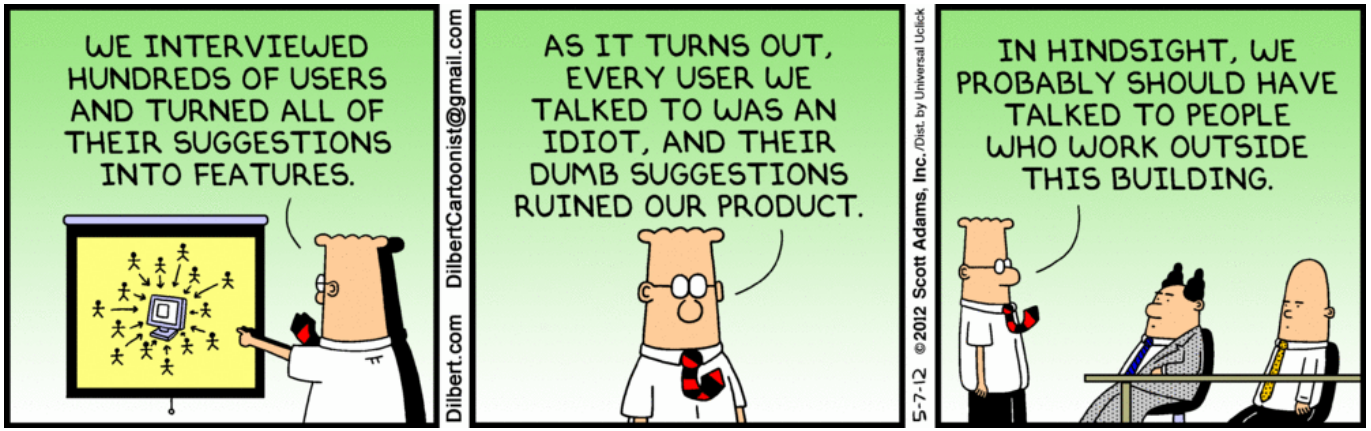
How can we elicit, express, verity and validate what the customer really needs? See Module 3 Right Product – Client Needs and Software Requirements



Usability

- Human factors, aesthetics, consistency, documentation, responsiveness
  - How effective is the product from the standpoint of the person who must use it?
  - Is it aesthetically acceptable?
  - Is the documentation accurate and complete?

How can we better understand user? See Module 3 Topic 3.2 User Interaction and Topic 3.5 Validating Requirements



Reliability

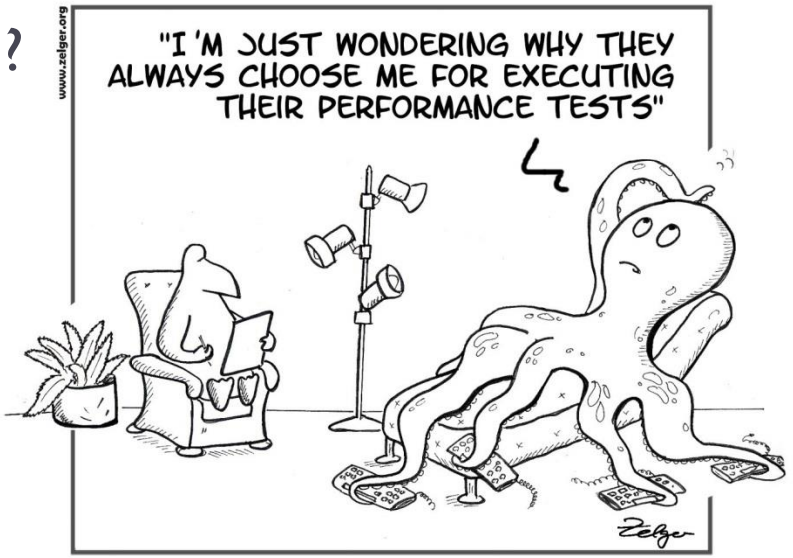
- Availability, Failure extent & Time length, Predictablity, Accuracy
  - What is the maximum acceptable system downtime?
  - Are failures predictable?
  - Can we demonstrate the accuracy of results?
  - How is the system recovered?

Can we reduce the risk of failure to an acceptable level? See Module 4 Done Right - Reviews, Testing and Metrics for Software Improvement

Performance

- Speed, efficiency, resource consumption, throughput, capacity, scalability
  - How fast must it be?
  - What's the maximum response time?
  - What's the throughput?
  - What's the memory consumption?

This is out of scope of this course

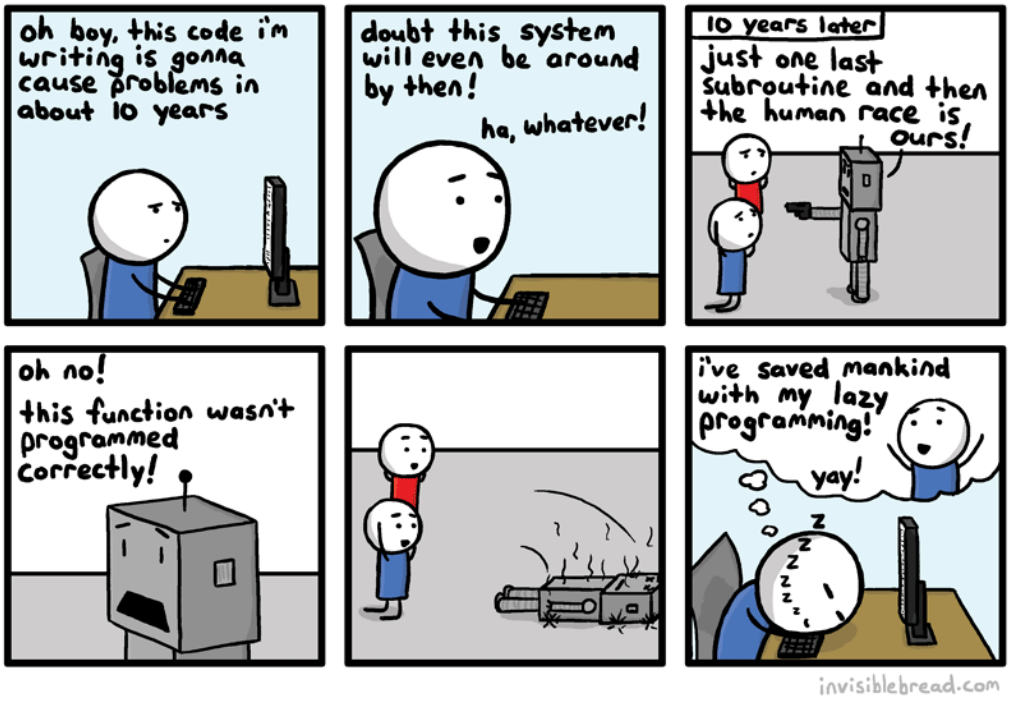


The multi-threaded octopus was looking for a change

Supportability

- Testability, Flexibility, Installability, Localizability
  - Is it testable, extensible, serviceable, installable, and configurable?
  - Can it be monitored?

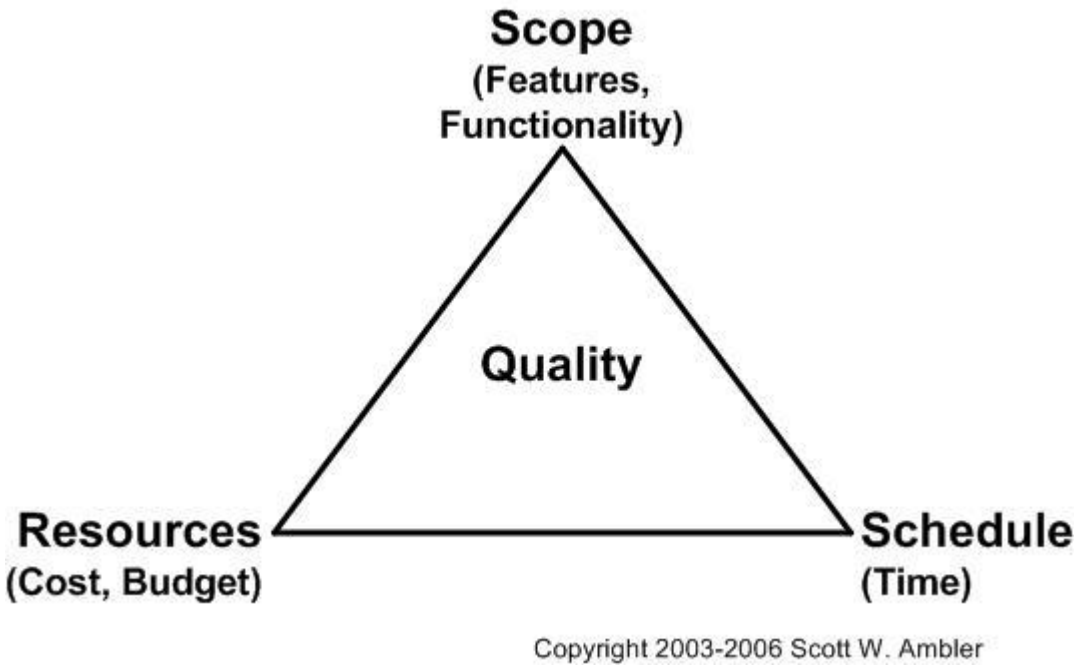
S is mainly for developers, testers and maintainers. See Module 4 Done Right - Reviews, Testing and Metrics for Software Improvement



Plus

- Design constraints
  - Do things like I/O devices or DBMS constrain how the software must be built?
- Implementation requirements
  - Is the use of TDD required?
  - Is statistically sound testing in the context of Cleanroom required?
- Interface requirements
  - What downstream feeds must be created? How frequent are feeds produced?
  - What other systems must this one interface with?
- Physical requirements
  - Must we be able to deploy to a machine no larger than 12" square, to be stationed in the Iraqi desert?

The "Iron" Triangle of Software Development



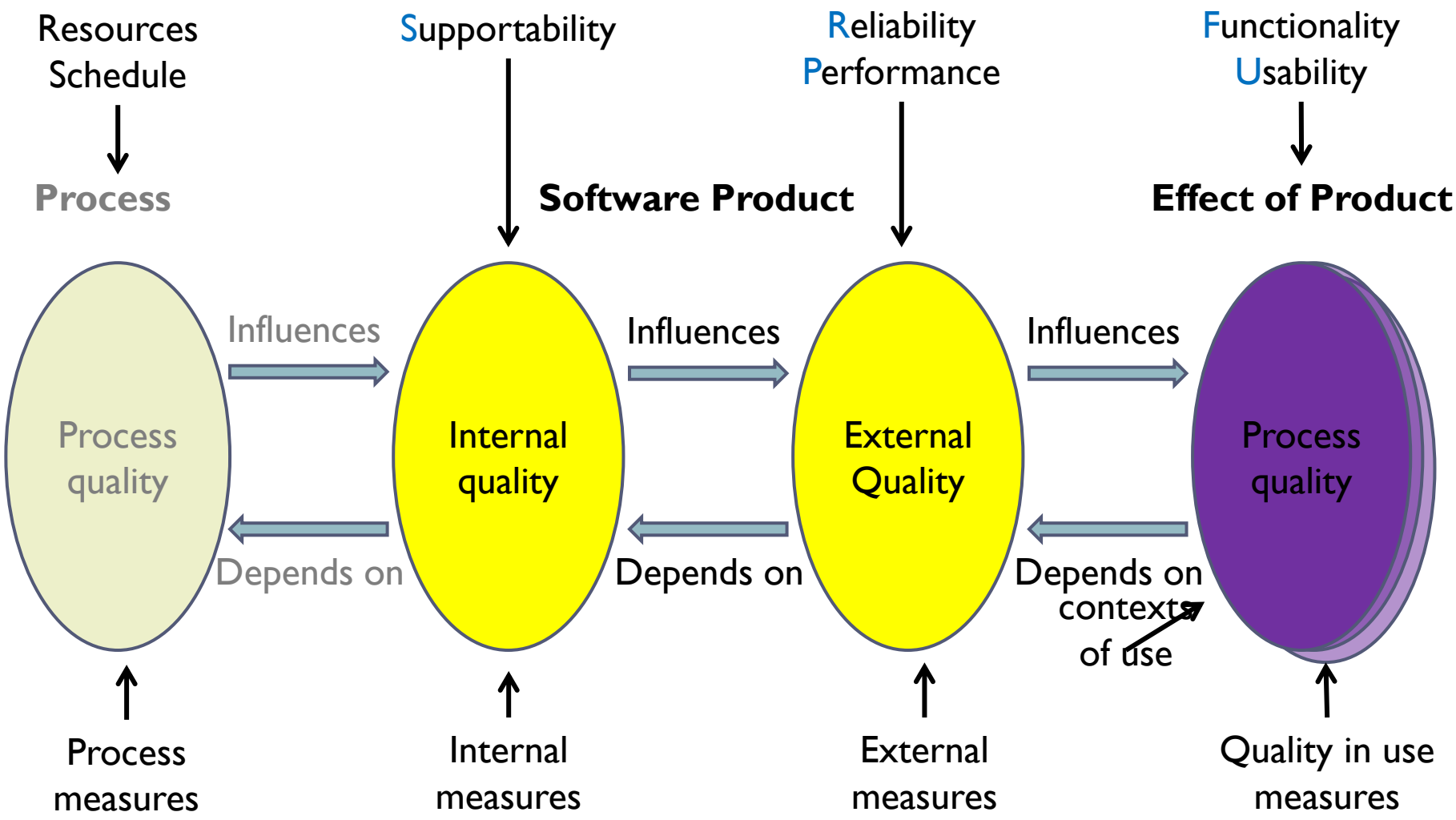
- Stakeholders have a different, and often conflicting, set of priorities
  - End users - Scope
  - Developers – Product quality
  - Business people – Resources
  - Senior management – Schedule

How to organize the work of people involved in the project? See Module 2 Software Process and Agile Practices and Module 5 Managed Right – Project Planning and Monitoring

Quality in the LifeCycle

Right Product, Done Right, Managed Right

Quality in the Lifecycle - ISO/IEC 9126 -1



Why Process?

Who you think typically compose a software development team?

A. Programmers, coders or developers
B. Clients
C. User interface specialists and graphic designers
D. Quality assurance specialists or testers

How Should They be Organized?

- Development team
  - What if there are many team members to coordinate?
  - How do we deal with huge differences between the productivity of one developer versus another?
  - What practices can teams follow to make the work more disciplined and less ad hoc?
  - What if there are lots of features to develop?
  - What if the project is intended to last several years?

- Customer
  - What does a customer really want in a software product?
  - What about its quality? What do they expect?
  - What if their needs change over time?

Software Process and Agile Practices – Module 2

- It takes the sustained work of many people with different skill sets to make great software products

- A process organizes the work on a product through its lifetime
  - Where you should start
  - Complete things in a logical order
  - Steps are not missed or overlooked

- A process also lays out rules and responsibilities for everyone on the development team

Why We Need Requirements?

- You are the product manager for a social network app. You get a seemingly simple request for your client

  "Users can talk to each other through the app"

  Is this clear? Will it be interpreted in different way?

Avoid confusion

Why We Need Requirements?

- One of your developers has begun developing the social network app and has come across this requirement

  "Users must be able to send text messages as a guest"

  What could go wrong?

Detect potential errors in your product before its' even built

Why Planning?

- Avoid overestimate or underestimate
- Avoid over commit to the client
- Reduce risk and plan ahead
  - What if a developer gets sick?
  - What is someone quits?
  - What if your client contact changes?
  - What if your technology crashes?

This is the focus of COMP3120 Managing Software Development!

Planning is not just for the beginning of a project!

Three Viewpoints (Goals) of Software Development

- Right product – Client Needs and Software Requirements (Module 3)
  - It meets client needs. It is easy to learn and easy to use. It does not waste their time. It looks nice. …
- Done right – Review, Testing and Metrics for Software Improvement (Module 4)
  - The software conforms to a specific design. And in turn, the design satisfies the stated set of requirements.
- Managed right - Project Planning and Monitoring (Module 5)
  - Set up just enough process and suitable practices to organize the work of everyone involved

Quick Question

The client produces a napkin, on which he has sketched out an app with a few outlined features. He says, this is the product we want. If you accept this document as your final set of requirements, rather than refining them with your client, which of the following are potential risks to your project?

A. Product implementation errors may not show up until software coding is finished and tested
B. The product may not satisfy the end users
C. The client has too much input over the look and feel of the product
D. The people involved are not collaborative, and morale drops.

Quick Question

Which term do we use to describe when a software product meets all the specified requirements?

A. Verified
B. Validated
C. Specified
D. Complete

Quick Question

You are testing your software product by having various users use the product. You are trying to determine if they like the user interface and user experience.

What is this an example of?

A. Verification
B. Validation

Quick Question

You are the product manager for a project that is developing a new video game. Your team is responsible for the character selection screen.

Which of these tasks would be in the verification and validation phase?

A. Writing tests for selecting a character
B. Planning what the characters will look like
C. Writing the source code for multiplayer selection
D. Executing tests for changing the colour of characters?

Monitoring, Metrics and Feedback

Why is Monitoring Important?

Please select a problem Scott will solve through monitoring.

A. Adapting to changing product requirements
B. Avoiding having to fire his least productive developer
C. Working without interruptions from the client
D. Meeting project plan deadlines

Monitoring, Metrics and Feedback

- Monitoring is essential for ensuring you are delivering the right product, done right and managed right
  - Module 3 Right Product – Topic 3.5 Validating Requirements
  - Module 4 Done Right – Topic 4.3 Software Metrics
  - Module 5 Managed Right – Topic 5.2 Monitoring Project Plan
- Metrics are an effective way of monitoring that gets quantifiable results
- Monitoring and metrics help to provide feedback that suggests improvements or affirms that you are on the right track

Why Agile?

The Agile Manifesto

Individuals & Interactions	over	Process & tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

Quick Question

Every two weeks, she invites her client to a Friday afternoon meeting where the development team demonstrates the latest product prototype. This week's meeting resulted in the client and development team discussing how a new user interface feature would allow users to easily enter information into the database. Paula documented the new requirements and made a note to herself to schedule a meeting with the legal team and the client. Signing-off on a revised contract is required, so the team has a specification of the new feature in writing.

Check all the ways Paula is in alignment with the values of the Agile Manifesto

A. A new prototype is demonstrated on a regular basis
B. Development team has face-to-face time with the client
C. Product evolves as new needs are identified
D. Contract is negotiated to specify the new feature

Career Opportunities

Course Aims and Values

Develop a holistic view of software development

Courses	Right Product	Done Right	Managed Right	Monitoring
What You Have Learned				
Programming Language & Methodology (e.g., COMP1140, 2100, 2140)		***		
Software Analysis and Design (e.g, COMP2130, 3530)	*	***		
Project Management (e.g., 3120)	*		***	
What You Need in Practice				
Software Engineering Project (e.g., COMP3500)	**	*****		
TechLauncher (e.g., COMP4500)	*****	*****	*****	***
What You will Learn				
Software Quality and Process (e.g., COMP4130)	User interaction Req'ts validation	Code reviews Testing	Software process Managing plan	Monitoring Metrics Feedback

Adds a set of new skills that are crucial for building better software

Career Opportunities

- The higher level you go, the more capable you are expected to be of monitoring your projects to ensure
  - they align to client needs (right product)
  - adhere to project plans (managed right)
  - achieve a desired level of quality (done right)

	Right Product	Done Right (Tech-wise)	Managed Right	Monitoring
Programmer		****		
Team Leader	*	*****	**	*
Project Manager	***	****	*****	***
Product Manager	*****	***	*****	*****