CMPT 473
Software Quality Assurance

Test Planning

Nick Sumner
Planning Tests

- We have looked at fundamental criteria for evaluating test suites.
Planning Tests

- We have looked at fundamental criteria for evaluating test suites.
- We still haven't tried to rigorously create tests.
Planning Tests

- We have looked at fundamental criteria for evaluating test suites.
- We still haven't tried to rigorously create tests.
- Recall: Quality comes from process
  - The testing process is guided by a *test plan*
Test Plans

**Test Plan**

- Documentation of the goals, concerns, methodology, and metrics for testing
Test Plans

**Test Plan**

- Documentation of the goals, concerns, methodology, and metrics for testing
- Not a test suite
Test Plans

Test Plan

- Documentation of the goals, concerns, methodology, and metrics for testing
- Not a test suite
- A process for applying and evaluating testing over the lifetime of a project
Test Plans

Test Plan

- Documentation of the goals, concerns, methodology, and metrics for testing
- Not a test suite
- A process for applying and evaluating testing over the lifetime of a project

What test plans have you seen so far in co-ops/internships?
What guidelines/rules did they contain?
Test Plans

**Test Plan**

- Documentation of the goals, concerns, methodology, and metrics for testing
- Not a test suite
- A process for applying and evaluating testing over the lifetime of a project

What test plans have you seen so far in co-ops/internships?
What guidelines/rules did they contain?

Were these useful? Why?
Test Plans

Test Plan

- Documentation of the goals, concerns, methodology, and metrics for testing
- Not a test suite
- A process for applying and evaluating testing over the lifetime of a project

Being too specific or detailed is a problem, the plan should guide the process but not become a burden itself.
Test Plans

Test Plan

- Documentation of the goals, concerns, methodology, and metrics for testing
- Not a test suite
- A process for applying and evaluating testing over the lifetime of a project

Being too specific or detailed is a problem, the plan should guide the process but not become a burden itself.

The nature & uses of a test plan depend on the project.
Why Use Test Plans?

- They act like requirements and specifications, but for the testing process
Why Use Test Plans?

- They act like requirements and specifications, but for the testing process

What happens if you skip clear requirements and specifications for the software itself?
Why Use Test Plans?

- They act like requirements and specifications, but for the testing process
  - Clear definition of what to do, how, and when
Why Use Test Plans?

- They act like requirements and specifications, but for the testing process
  - Clear definition of what to do, how, and when
  - Enable developers (& customers) to communicate about the process
Why Use Test Plans?

- They act like requirements and specifications, but for the testing process
  - Clear definition of what to do, how, and when
  - Enable developers (& customers) to communicate about the process
  - Structures the process to aid in scheduling & management
Why Use Test Plans?

- They act like requirements and specifications, but for the testing process
  - Clear definition of what to do, how, and when
  - Enable developers (& customers) to communicate about the process
  - Structures the process to aid in scheduling & management
  - Testing after the fact is too late
What Should It Include?

Structure may vary but contents often show

• Purpose & Objectives
What Should It Include?

Structure may vary but contents often show

- Purpose & Objectives
- Requirements
  - Environment, resources, dependencies
What Should It Include?

Structure may vary but contents often show

- Purpose & Objectives
- Requirements
  - Environment, resources, dependencies
- Schedule and/or personal responsibilities
What Should It Include?

Structure may vary but contents often show:

- Purpose & Objectives
- Requirements
  - Environment, resources, dependencies
- Schedule and/or personal responsibilities
- Evaluation Criteria (Test Requirements)
What Should It Include?

Structure may vary but contents often show

- Purpose & Objectives
- Requirements
  - Environment, resources, dependencies
- Schedule and/or personal responsibilities
- Evaluation Criteria (Test Requirements)
- Expected Risks & Contingencies
What Should It Include?

Structure may vary but contents often show

- Purpose & Objectives
- Requirements
  - Environment, resources, dependencies
- Schedule and/or personal responsibilities
- Evaluation Criteria (Test Requirements)
- Expected Risks & Contingencies
- Deliverables
Purpose

May apply testing at many phases of the lifecycle:

- *Unit test plans*- correctness of individual components
Purpose

May apply testing at many phases of the lifecycle:

- *Unit test plans*- correctness of individual components
- *Integration test plans*- correct component interaction
Purpose

May apply testing at many phases of the lifecycle:

- *Unit test plans* - correctness of individual components
- *Integration test plans* - correct component interaction
- *System test plans* - whole system concerns like correctness, throughput, responsiveness, & reliability
May apply testing at many phases of the lifecycle:

- **Unit test plans**- correctness of individual components
- **Integration test plans**- correct component interaction
- **System test plans**- whole system concerns like correctness, throughput, responsiveness, & reliability
- **Acceptance test plans**- adherence to customer requirements
Purpose

May apply testing at many phases of the lifecycle:

- **Unit test plans** - correctness of individual components
- **Integration test plans** - correct component interaction
- **System test plans** - whole system concerns like correctness, throughput, responsiveness, & reliability
- **Acceptance test plans** - adherence to customer requirements
- **Regression test plans** - regular tests to maintain quality
Purpose

May apply testing at many phases of the lifecycle:

- *Unit test plans*- correctness of individual components
- *Integration test plans*- correct component interaction
- *System test plans*- whole system concerns like correctness, throughput, responsiveness, & reliability
- *Acceptance test plans*- adherence to customer requirements
- *Regression test plans*- regular tests to maintain quality
- *Master test plans*- overall plan for all testing
- ...
Objectives

Clearly identify the goals of testing
Objectives

Clearly identify the goals of testing

- Specification of the software & feature under test
Objectives

Clearly identify the goals of testing

- Specification of the software & feature under test
- Functional requirements that the test process should help assure.
Objectives

Clearly identify the goals of testing

- Specification of the software & feature under test
- Functional requirements that the test process should help assure.
- Nonfunctional requirements
  - resources, performance, reliability, portability, usability, ...


Test Process Requirements

Environmental

- What platforms, resources, and other preconditions are assumed for running the test?
Test Process Requirements

Environmental

• What platforms, resources, and other preconditions are assumed for running the test?
  – e.g. operating system, operating system version, supporting software/libraries, build tools, processor, attached hardware, remote networks/servers, ...
Addressing Multiple Configurations

Many different environments & build options may be desired.
Addressing Multiple Configurations

Many different environments & build options may be desired.

- Once again, combinatorial complexities arise
Addressing Multiple Configurations

Many different environments & build options may be desired.

- Once again, combinatorial complexities arise

Options

- Test most expected / common configurations
Addressing Multiple Configurations

Many different environments & build options may be desired.

- Once again, combinatorial complexities arise

Options

- Test most expected / common configurations
- Test most extreme scenarios
Addressing Multiple Configurations

Many different environments & build options may be desired.

- Once again, combinatorial complexities arise

Options

- Test most expected / common configurations
- Test most extreme scenarios
- Combinatorial testing once again
Deliverables

What are the intended results?
Deliverables

What are the intended results?

What do you expect to see from running a test suite?
Deliverables

What are the intended results?

- Documented test cases?
- Test scripts?
- Logs of the test process?
- Summaries of failures?
- Aggregate summaries? (How many failed/passed)
- Adequacy assessments?
Balancing

Test plans can be *either* an asset or a liability.
Balancing

Test plans can be *either* an asset or a liability.

- Monolithic & high risk process?
Balancing

Test plans can be *either* an asset or a liability.

- Monolithic & high risk process?
  - Lasting detailed documentation helps mitigate risk
Balancing

Test plans can be *either* an asset or a liability.

- Monolithic & high risk process?
  - Lasting detailed documentation helps mitigate risk
- Fast paced, agile, & low risk?
Balancing

Test plans can be *either* an asset or a liability.

- Monolithic & high risk process?
  - Lasting detailed documentation helps mitigate risk
- Fast paced, agile, & low risk?
  - Less detailed docs & more detailed whiteboarding.
Balancing

Test plans can be *either* an asset or a liability.

- Monolithic & high risk process?
  - Lasting detailed documentation helps mitigate risk
- Fast paced, agile, & low risk?
  - Less detailed docs & more detailed whiteboarding.

There is no cookie cutter process to follow *every* time.
Contestation

- There is a growing tendency against traditional test plans
  - Why might this be the case?
Contention

- There is a growing tendency against traditional test plans
  - Why might this be the case?
- Lighter weight approaches examine higher level project quality
  - Easier to visualize & manage
Contention

● There is a growing tendency against traditional test plans
  – Why might this be the case?

● Lighter weight approaches examine higher level project quality
  – Easier to visualize & manage
  – Don't miss the forest for the trees
Contention

• There is a growing tendency against traditional test plans
  – Why might this be the case?

• Lighter weight approaches examine higher level project quality
  – Easier to visualize & manage
  – Don't miss the forest for the trees
  – Don't require updating project level documents
Attribute-Component-Capability

- **Attributes**
  - *(adj)* high level (non)functional properties to ensure
Attribute-Component-Capability

- **Attributes**
  - *(adj)* high level (non)functional properties to ensure

- **Components**
  - *(noun)* chunks of software or features
Attribute-Component-Capability

- **Attributes**
  - *(adj)* high level (non)functional properties to ensure

- **Components**
  - *(noun)* chunks of software or features

- **Capabilities**
  - *(verb)* user actions
Attribute-Component-Capability

- **Attributes**
  - *(adj)* high level (non)functional properties to ensure
- **Components**
  - *(noun)* chunks of software or features
- **Capabilities**
  - *(verb)* user actions

Test process is organized by a matrix showing how attributes are satisfied
## ACC Matrix

Attributes & Components form the columns & rows of a matrix.

<table>
<thead>
<tr>
<th>Component</th>
<th>Fast</th>
<th>Secure</th>
<th>Personalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shopping Cart</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item Search</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notifications</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## ACC Matrix

<table>
<thead>
<tr>
<th>Component</th>
<th>Fast</th>
<th>Secure</th>
<th>Personalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shopping Cart</td>
<td></td>
<td>• Credit card information is stored encrypted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Passwords are not stored plain-text</td>
<td></td>
</tr>
<tr>
<td>Item Search</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notifications</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Capabilities for each component fit into the cells.
## ACC Matrix

<table>
<thead>
<tr>
<th>Component</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fast</td>
</tr>
<tr>
<td>Shopping Cart</td>
<td></td>
</tr>
<tr>
<td>Item Search</td>
<td></td>
</tr>
<tr>
<td>Notifications</td>
<td></td>
</tr>
</tbody>
</table>

Test cases fall into cells of the matrix.
## ACC Matrix

<table>
<thead>
<tr>
<th>Component</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fast</td>
</tr>
<tr>
<td>Shopping Cart</td>
<td>Green</td>
</tr>
<tr>
<td>Item Search</td>
<td></td>
</tr>
<tr>
<td>Notifications</td>
<td>Green</td>
</tr>
</tbody>
</table>

Results can be easily visualized.
ACC Testing

- What are the benefits of this approach?
ACC Testing

- What are the benefits of this approach?
- What are the risks?
A traditional test plan formally specifies the goals, assumptions, methodologies, & outcomes.
Summary

- A traditional test plan formally specifies the goals, assumptions, methodologies, & outcomes.
- Newer test plans take higher level perspectives.
Summary

- A traditional test plan formally specifies the goals, assumptions, methodologies, & outcomes.
- Newer test plans take higher level perspectives.
- Neither approach is best in general.

Once again, there is no silver bullet.