

OFF THE HAPPY PATH

LOOKING AT REQUIREMENTS FROM AN EXPLORATORY TESTING POINT OF
VIEW



PRESENTATION ABSTRACT

Let's face it, there are more things that you want the application NOT to do than to do. As a tester, it's my job to figure out how to gain access to all the unmarked paths surrounding the happy path and go on a bug hunt!

When going on a bug hunt, I follow each of the unmarked paths until I am stopped. Every path is a mystery. I wonder what I will find. Is this path here because of missing requirements, incomplete requirements or a developer's unique interpretation of the desired functionality?

Come exploring with me as I share the most common reasons for unmarked paths and tips on how to create requirements that keep your tester (and users) traveling down the happy path.



TEAM MEMBER FOCUS

- **Project Manager** focus is what is in scope.
- **Business Analysts** focus on determining and documenting what the customer wants to achieve based on what is in scope.
- **Customer** focus is to get their job done in the easiest way possible.
- **Developers** focus on creating a solution that bring the requirements to life while staying in the boundaries set by the scope.
- **Quality Assurance** focus on verifying that the requirements are all satisfied and application does not do anything unexpected.

POSITIVE: SOMETHING I WANT TO HAPPEN.



Picture Source: <http://www.crossingitaly.net/travel/wp-content/uploads/Beach-holidays.jpg>

HAPPY PATH

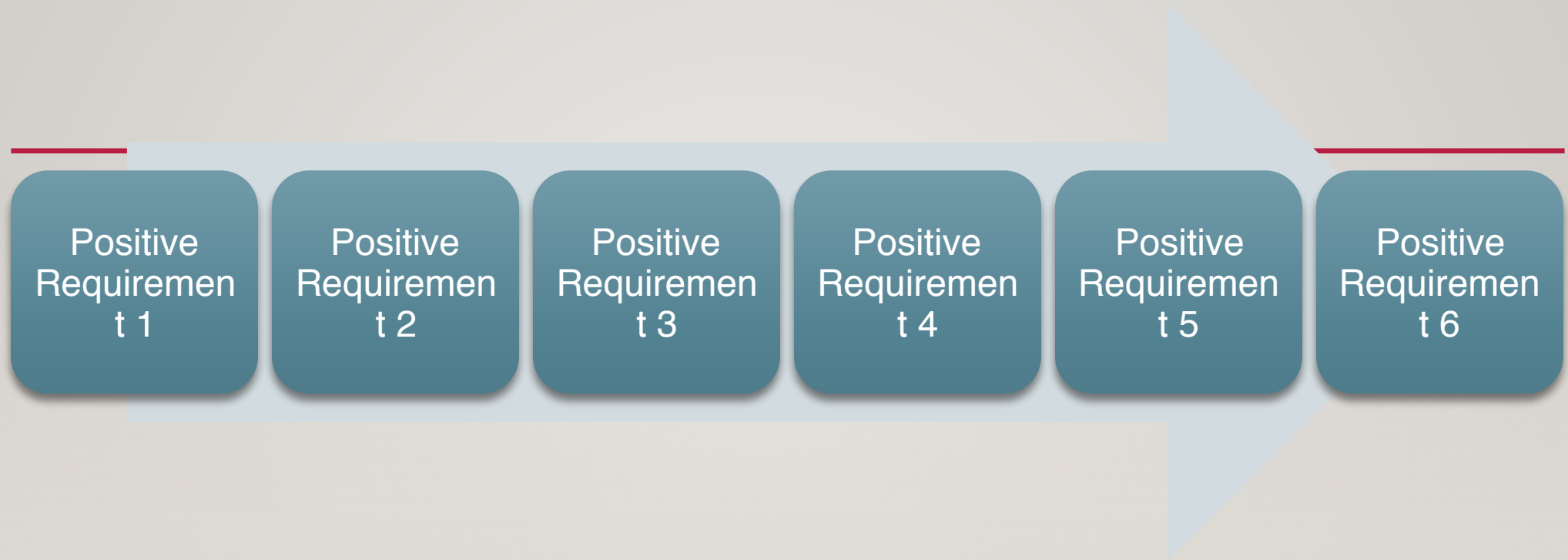
Happy path is meant to describe the ideal process, without any exceptions or alternate paths.

Architect Solutions The problem with the "Happy Path"



POSITIVE REQUIREMENTS

- Support the business by being present.
- Documented “What” the business needs/wants
 - Acceptance Criteria
 - The application SHALL/Must...
- Standards
 - Performance
 - Security
 - Regulatory



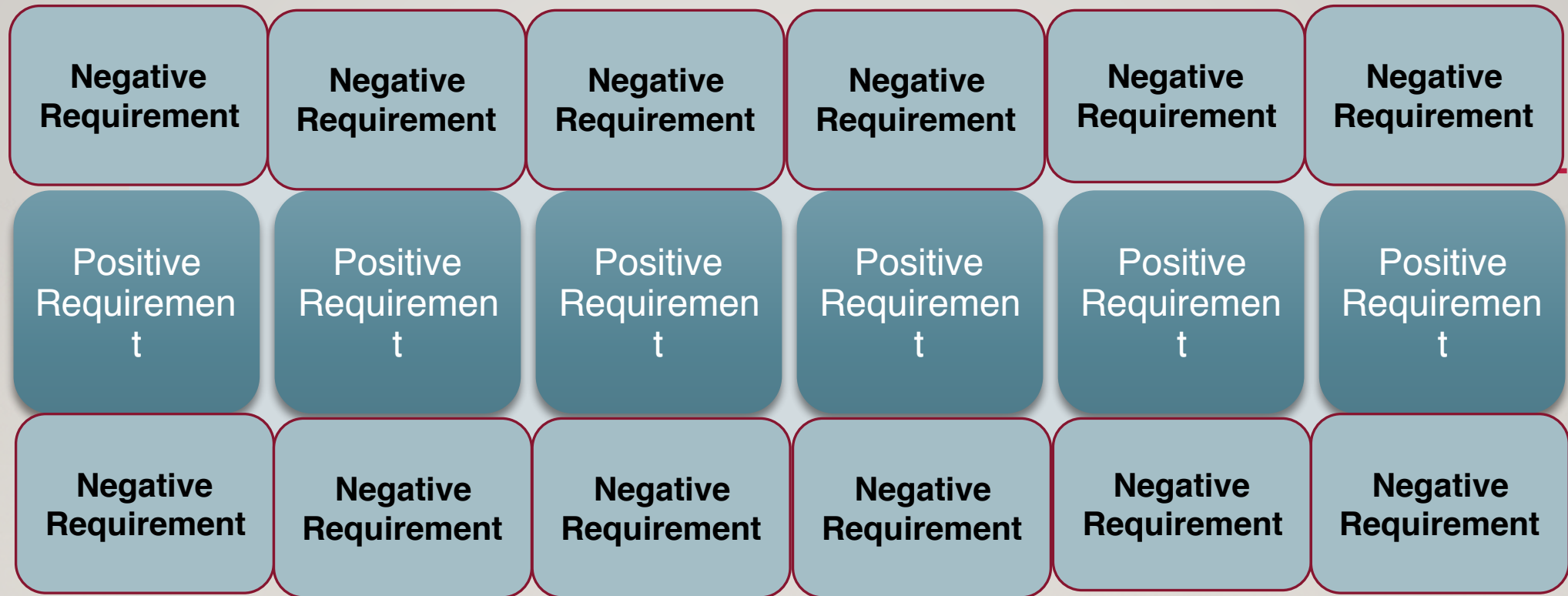
HOW TO BUILD A HAPPY PATH

NEGATIVE: SOMETHING I DON'T WANT TO HAPPEN



NEGATIVE REQUIREMENTS

- Supports the business by NOT being present.
- Implied “What” the business does not want
 - Non-Manager will NOT be able to...
 - The application SHALL NOT/MUST NOT...
- Violation of Standards
 - Application Crashes
 - Does not limit access to Private Information (PI)



NEGATIVE REQUIREMENTS AND THE HAPPY PATH

**“THE PRINCIPLE OBJECTIVE OF SOFTWARE TESTING
IS TO GIVE CONFIDENCE IN THE SOFTWARE.” – ANONYMOUS**





POSITIVE TESTING

Verify that the application does what is expected.



NEGATIVE TESTING

Verify That The
Application Doesn't Do
Anything Unexpected.

SCRIPTED TEST

Definition: Tests that are created based on requirements/standards where documentation is completed before testing begins.

Purpose: Test the known

- Test positive requirements
- Test negative requirements

Execution: Manual or Automated

SCRIPTED TEST EXAMPLE

Positive Requirement:

As a Manager, I want to be able to reassign a work item to the department director I report to, so that it can be escalated.

Positive Scripted Test:

Logged in as a manager, I can reassign a work item to my department director.

Negative Scripted Test:

- Logged in as a manager, I can not reassign a work item to a department director that I do not directly report to.
- Logged in as a manager, I can not reassign all my work items to my department director.
- Logged in as a manager, I can not reassign a work item to a manager.
- Logged in as a customer service rep, I can not reassign a work item to my department director.



“More than the act of testing, the act of designing tests is one of the best bug preventers known. The thinking that must be done to create a useful test can discover and eliminate bugs before they are coded – indeed, test-design thinking can discover and eliminate bugs at every stage in the creation of software, from conception to specification, to design, coding and the rest.” – Boris Beizer

EXPLORATORY TESTING

Definition: Manual testing that systematically explores the application to gain a deeper understanding of the functionality of the application around the boundaries of requirements/standards.

Purpose: Test the “space” between and around the positive and negative requirements/scripted test.

- Find less obvious tests that need documented.
- Identify missing requirements, creative interpretations of requirements or inconsistencies in functionalities.

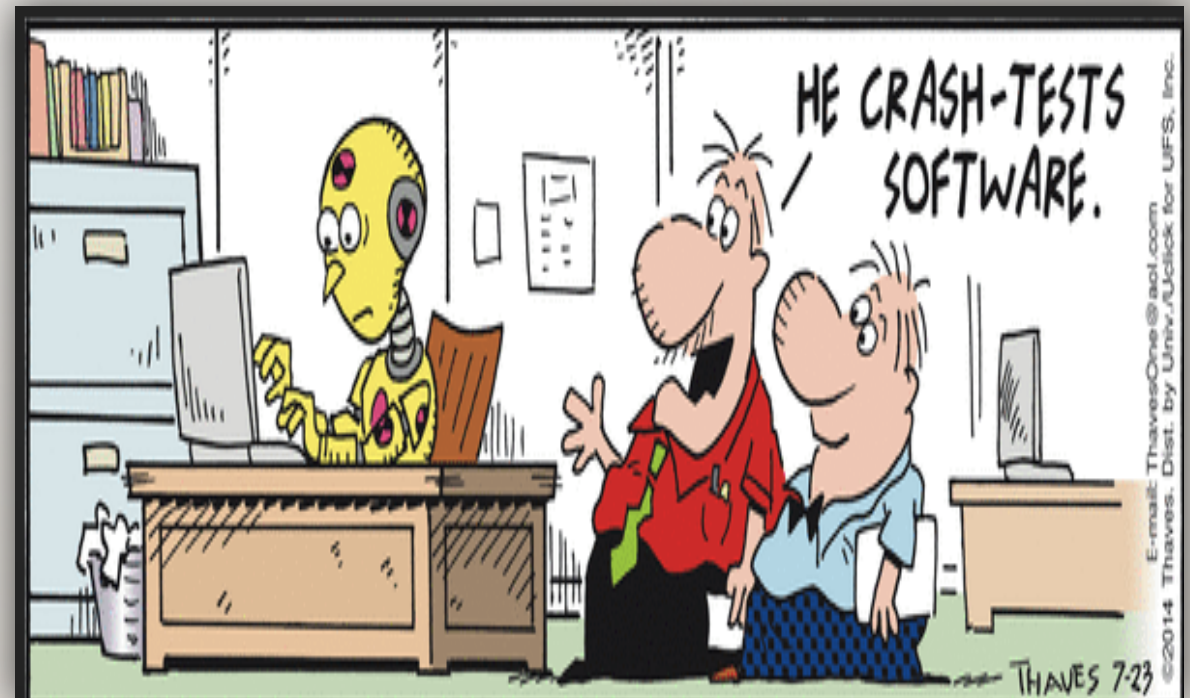
Execution: Manual

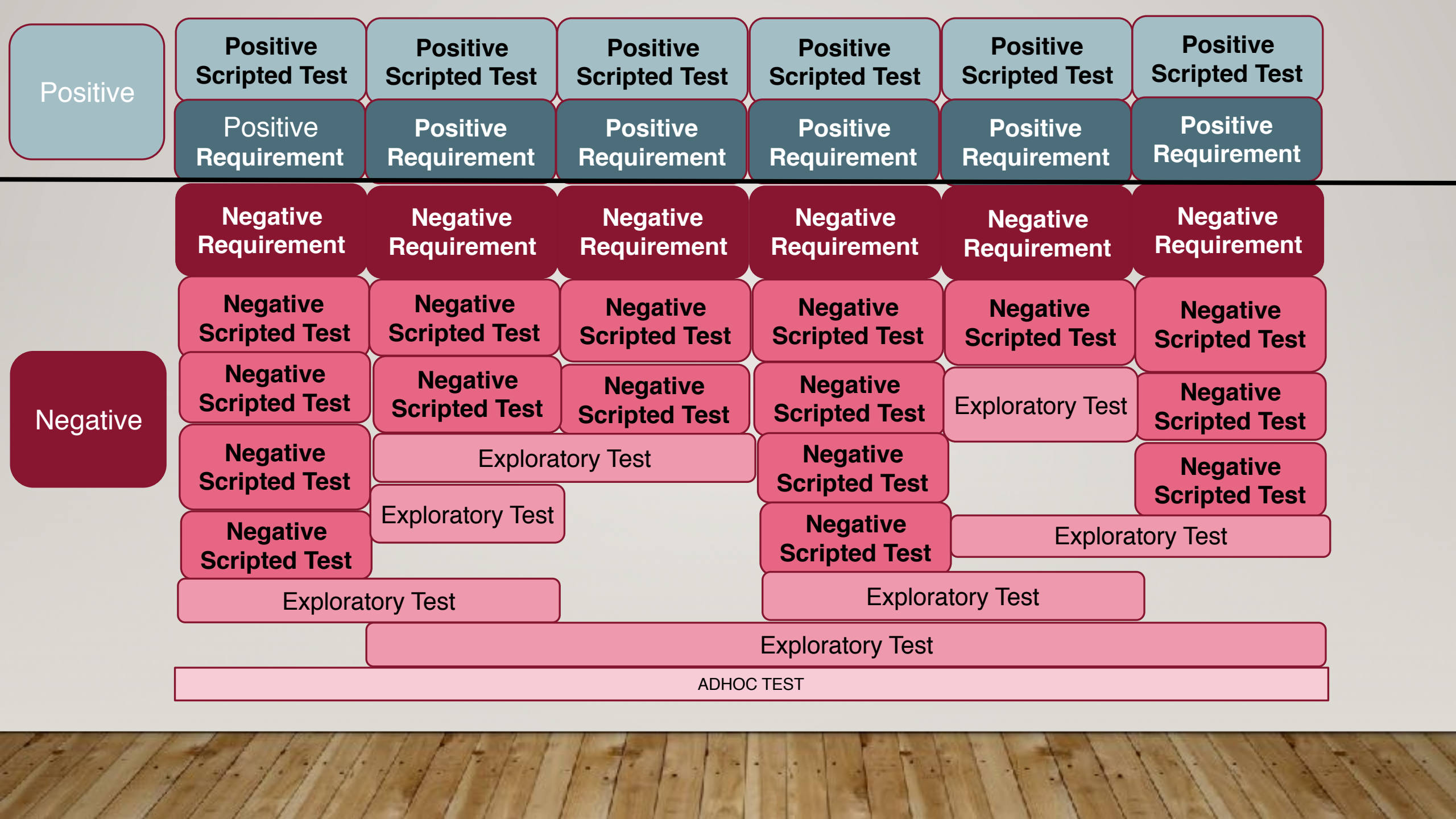
ADHOC TESTING (BUG HUNTING)

Definition: Free flowing manual testing that explores the application outside of the boundaries of requirements/ standards.

Purpose: Identify fringe cases and bugs that are off the expected user paths.

Execution: Manual





Positive

Positive
Scripted Test

Positive
Scripted Test

Positive
Scripted Test

Positive
Scripted Test

Positive
Scripted Test

Positive
Scripted Test

Positive
Requirement

Positive
Requirement

Positive
Requirement

Positive
Requirement

Positive
Requirement

Positive
Requirement

Negative
Requirement

Negative
Requirement

Negative
Requirement

Negative
Requirement

Negative
Requirement

Negative
Requirement

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Exploratory Test

Negative
Scripted Test

Negative
Scripted Test

Exploratory Test

Negative
Scripted Test

Negative
Scripted Test

Negative
Scripted Test

Exploratory Test

Negative
Scripted Test

Exploratory Test

Exploratory Test

Exploratory Test

Exploratory Test

ADHOC TEST

MARKETING CASE STUDY

Marketing wants to send out a targeted marketing letters to all active accounts for State of Confusion 457 plan that have not had a contribution in 6 months or more to encourage people to save more for retirement.

Positive Requirements:

- Identify all Great State of Confusion 457 accounts that have not had a contribution in six months or longer that are in Active status.
- Generate a mail file that contains: Account Type, Account Status, Name, Address, City, State, Zip, Date of last deposit, Gender

MARKETING CASE STUDY

Positive Scripted Tests:

- Great State of Confusion accounts with an Account Status of “A” and last contribution date that is exactly 6 months prior to current date is included in the file.
- Great State of Confusion accounts with an Account Status of “A” and last contribution is 6 months and 1 day prior to current date is included on the file.
- Mail file contains: Account Type, Account Status, Name, Address, City, State, Zip, Date of last deposit, Gender

MARKETING CASE STUDY

Negative Scripted Tests:

- Great State of Confusion accounts with an Account Status of “A” with last contribution date that is 5 months and 31 days prior to current date is not included in the file.
- Great State of Confusion accounts with an Account Status of “C” for closed are not in file.
- Great State of Confusion accounts with an Account Status of “P” for in payout are not included in the file.
- Non-Great State of Confusion accounts with an Account Status of “A” and last contribution date that is exactly 6 months prior to current date is included in the file.
- Non-Great State of Confusion accounts with an Account Status of “A” and last contribution of 6 months and 1 day prior to current date is included on the file.

MARKETING CASE STUDY

Exploratory Testing

Looking for answers to questions about functionality that are not covered by positive requirements.

- What about accounts that have pending activity such as contributions, distributions or loans?
- What about account identified for Escheatment (unclaimed funds)?
- What about accounts that have a known bad address?
- What about accounts that have a hold reason code?

MARKETING CASE STUDY

Exploratory testing discovered missing requirements that stopped the mailing of marketing material to people who were not eligible to make contributions.

Reason Code indicates if there is a hold on the account.

- A = Active
- D = Deceased
- L = Legal Hold
- H = General hold on account
- M = Military Deployment
- S = Separation from employment
- No code = Active

TERMS AND DEFINITIONS

- **Positive Requirements:** Supports the business by being present.
- **Negative Requirements:** Supports the business by NOT being present.
- **Positive Scripted Tests:** Documented tests that test Positive Requirements
- **Negative Scripted Tests:** Documented tests that test Negative Requirements
- **Exploratory Testing:** Manual testing that systematically explores the application to gain a deeper understanding of the functionality of the application around the boundaries of requirements/standards.
- **ADHOC Testing:** Free flowing manual testing that explores the application outside of the boundaries of requirements/standards.

AGILE AND HAPPY PATH

The reality is that “Agile” is not “Anti-Waterfall”, “Agile” is “Anti-Happy Path”. Agile is about recognizing that at the start of a project we are at our moment of maximum ignorance. It’s about embracing the fact that the act of developing software shines a light into the domain and uncovers hidden corners and edges. It’s about realizing that trips off the happy path aren’t exceptions, the happy path is the exception.

Richard Dalton <http://www.devjoy.com/2014/02/the-happy-path-model-of-software-development/>



PRESENTER PROFILE

Cindy Hufnagle is the lead IT Software QA Engineer for WillowWood. She has 19 years of experience working in Information Technology the roles of business analyst, software quality assurance, project manager, risk analyst, data management analyst, technical trainer and technical writer.

Linkedin:<https://www.linkedin.com/in/cindy-hufnagle-pmp-mba-psm-1-61878811>

Email: cynhufnagle@gmail.com

REFERENCES

- **Adrian Reed:** Debate: The Danger of "Negative" Requirements –**August 24, 2012** <https://www.techwell.com/techwell-insights/2012/08/debate-danger-negative-requirements>
- **Architect Solutions** -<http://www.architechsolutions.com/>
- **BA Times** - <https://www.batimes.com>
- **PM Times** - <https://www.pmtimes.com>
- Richard Dalton **The “Happy Path Model” of software development** <http://www.devjoy.com/2014/02/the-happy-path-model-of-software-development/>
- Sean M Everett **What is a “Happy Path”?** Thinking about user experience in an entirely new light <https://humanizing.tech/what-is-a-happy-path-2c62959b6d21>
- **TidalShift Inc** - <http://www.tidalshift.ca>