

# Agile + DevOps **WEST**

A TECHWELL EVENT

## **AD42**

Agile Testing & Test Automation

3:15 PM

## **AD42 - Reality-Driven Testing in Agile Projects**

Presented by:

**Robert Sabourin**

AmiBug.Com, Inc.

Brought to you by:



888-268-8770 · 904-278-0524 - [info@techwell.com](mailto:info@techwell.com) - <https://agiledevopswest.techwell.com/>

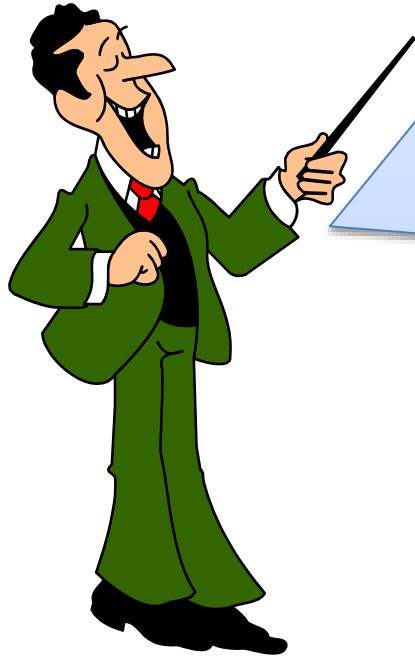
# Robert Sabourin

Rob Sabourin has more than thirty-five years of management experience leading teams of software development professionals. A highly-respected member of the software engineering community, Rob has managed, trained, mentored, and coached hundreds of top professionals in the field. He frequently speaks at conferences and writes on software engineering, SQA, testing, management, and internationalization. Rob authored *I am a Bug!*, the popular software testing children's book; works as an adjunct professor of software engineering at McGill University; and serves as the principal consultant (and president/janitor) of AmiBug.Com, Inc. Contact Rob at [rsabourin@amibug.com](mailto:rsabourin@amibug.com).



# **REALITY DRIVEN TESTING IN AGILE PROJECTS**

# Instructor Introduction



Presented By

Robert Sabourin  
robsab@gmail.com

# Reality Driven Testing



- Robert Sabourin ,  
*Software Evangelist*
- President
- AmiBug.Com Inc.
- Montreal, Quebec,  
Canada
- robsab@gmail.com

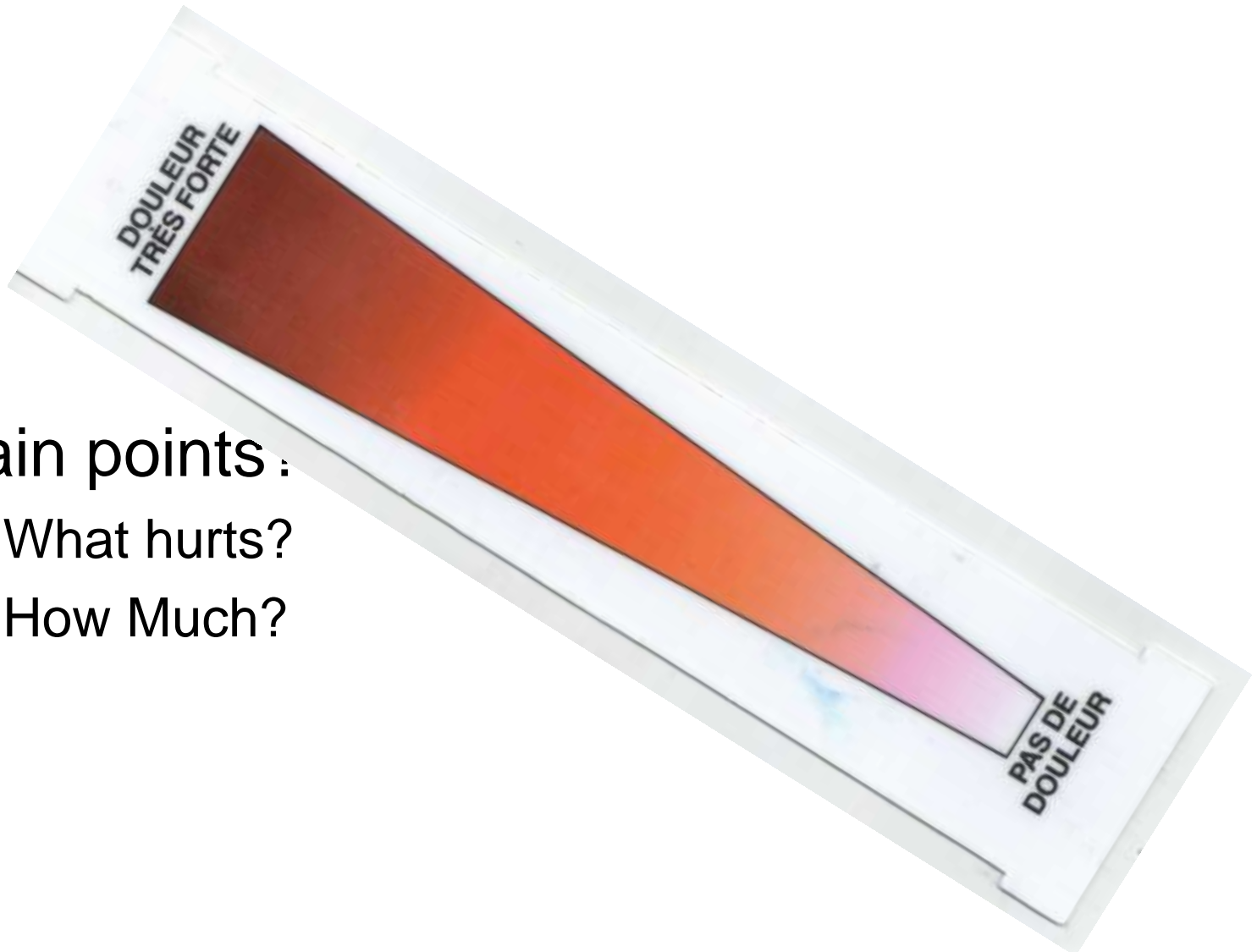


1

---

# **SOME PHILOSOPHY**

# Reality Driven Testing



- Pain points:
  - What hurts?
  - How Much?

# Fundamental Question

- How do you know when you are finished?





# Reality Driven Testing



What is Quality?



# Reality Driven Testing

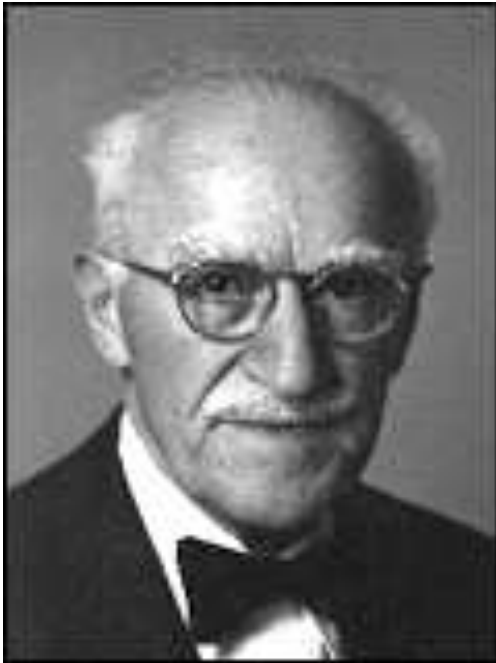
## Phil Crosby

- “Quality is defined as conformance to requirements”
- “Quality is not a measure of GOODNESS”
  - Phil B. Crosby, *Quality is Free*



# Reality Driven Testing

## Joseph Juran



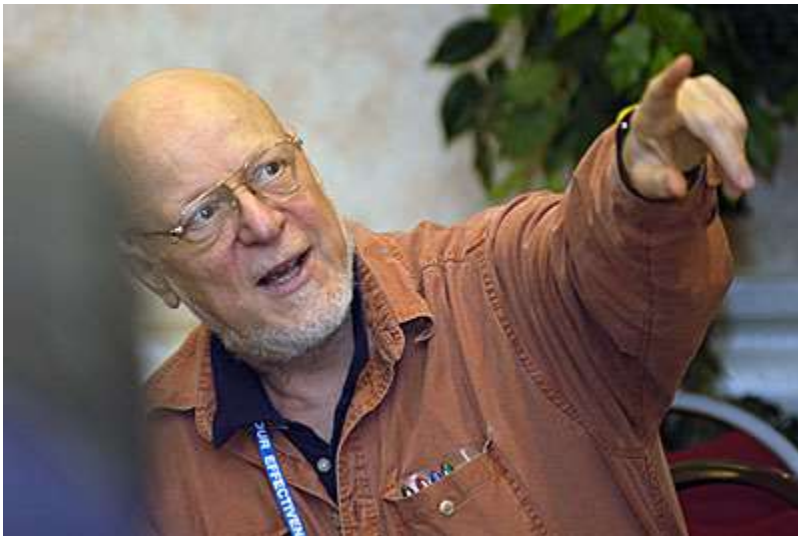
**“Quality is fitness for use”**

**Quality Control  
Handbook**

# Reality Driven Testing

## Gerald M. Weinberg

“Quality is value to some person”



**Exploring Requirements  
Quality Before Design**

**Dorset House**

# Reality Driven Testing



Conforming to requirements



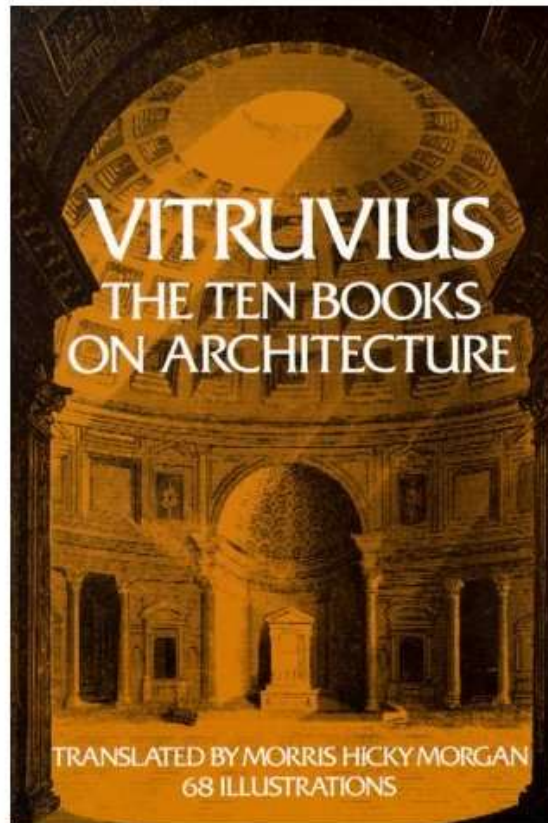
Suiting purpose



Delivering value to stakeholders

# Reality Driven Testing

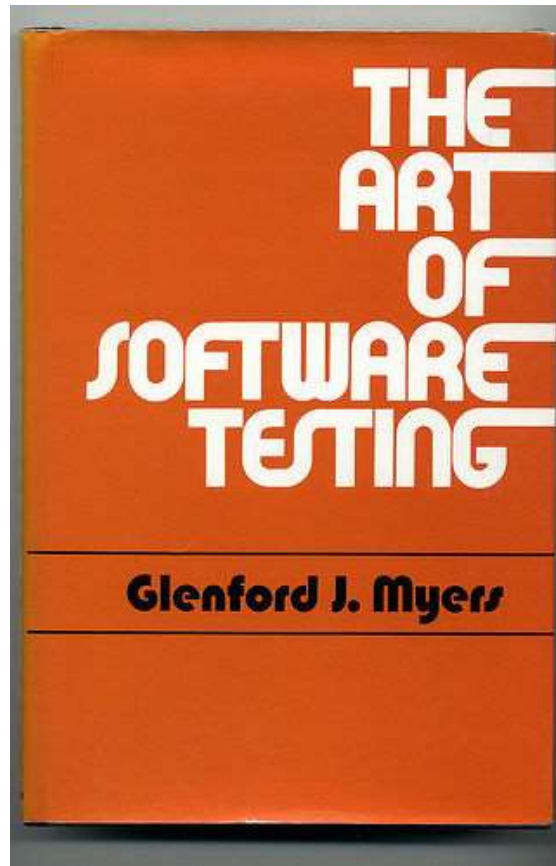
## Circa 25 BC



# Reality Driven Testing

---

**Circa 1979 AD**



# Reality Driven Testing

## Edsger W. Dijkstra

- “Program testing can be used to show the presence of bugs, but never to show their absence”





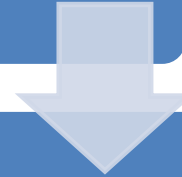
# Reality Driven Testing

---

Real changes

Real usage

Real world

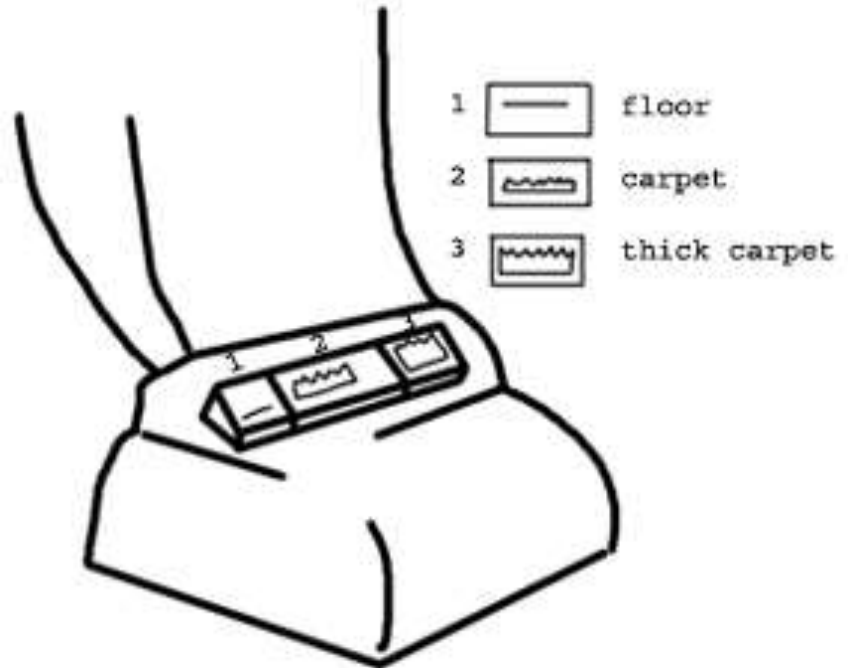




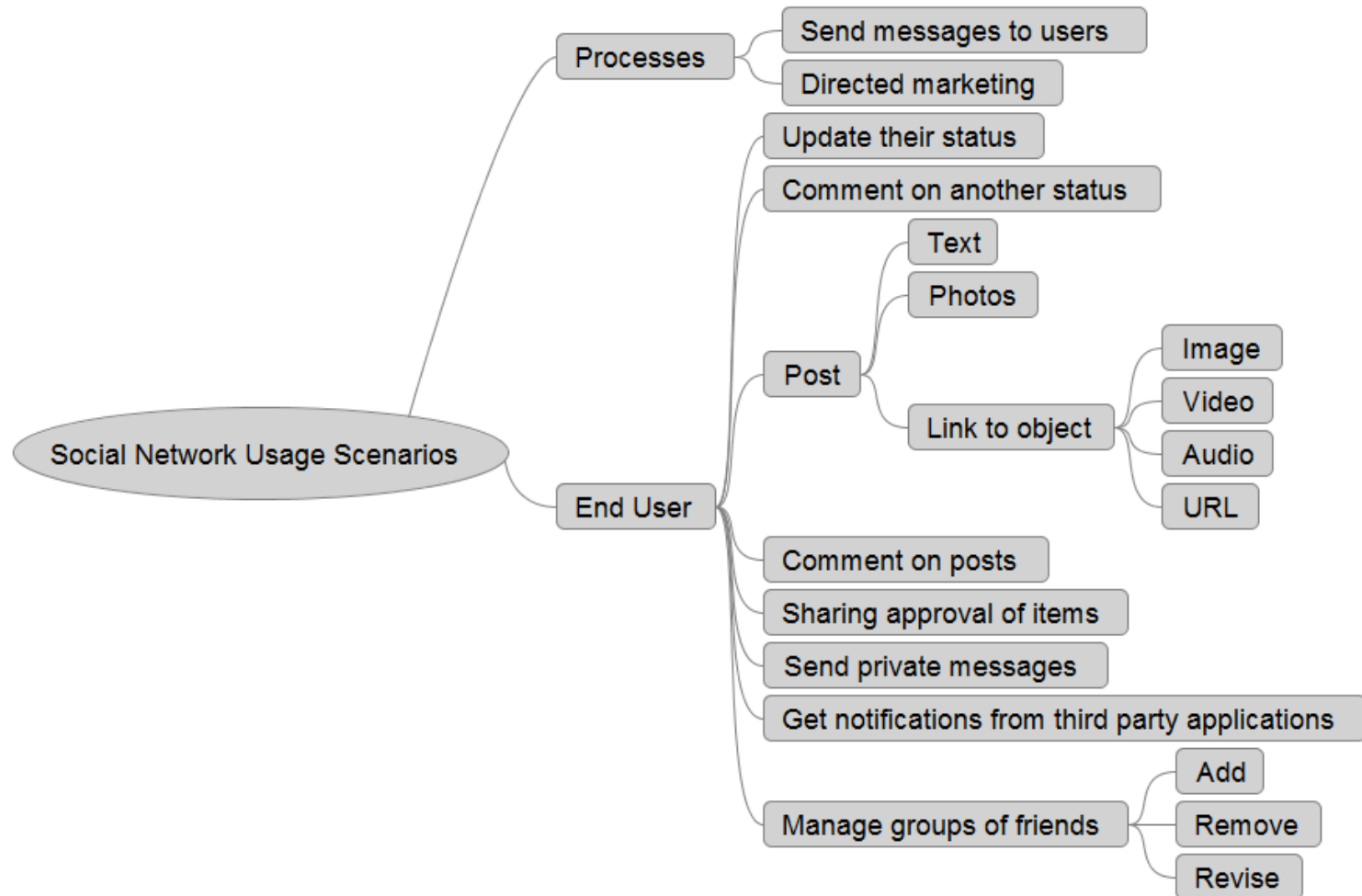
# Charter Types

## Usage Scenarios

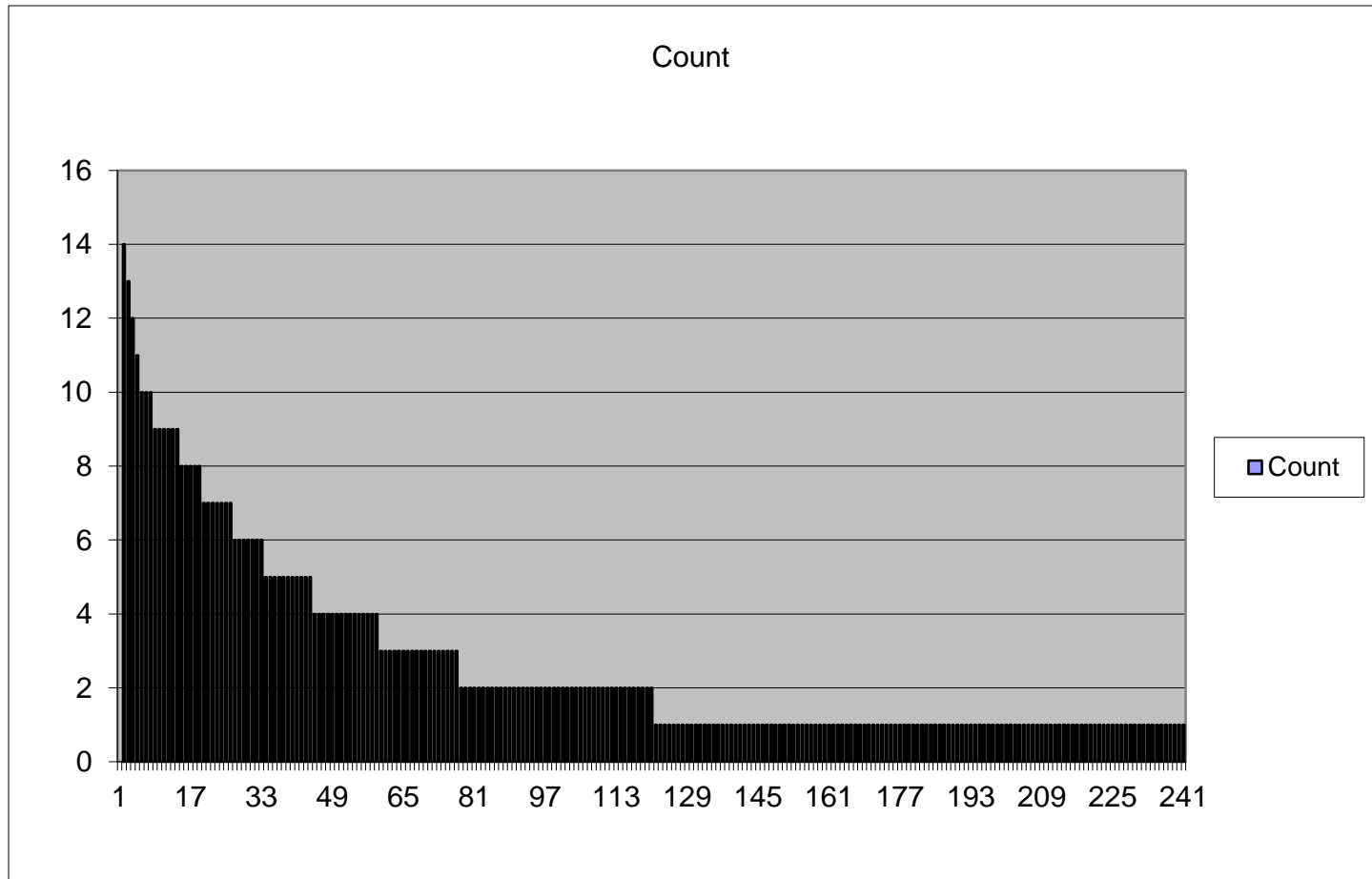
- Identify classes of users
- Identify how users will use system
- Describe scenarios
- Use Story board or similar approaches
- Identify variations



# Usage Scenarios



# Pareto Combinations



# Usage Scenarios

## Sharing a Picture With a Group of Friends



### Image Sources

- Different sources
- Local
- Camera
- Phone
- Memory card
- USB stick

# Usage Scenarios

## Sharing a Picture With a Group of Friends

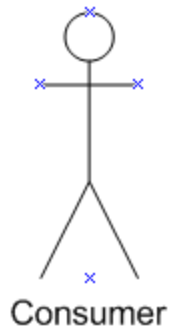


### Images

- Image formats
- Resolutions
- Sizes
- Colors
- None
- Many

# Usage Scenarios

## Sharing a Picture With a Group of Friends



Request available groups of friends



Friend Lists  
Manager

An oval actor representing a system component, labeled "Friend Lists Manager". It has a double-line border and small blue 'x' marks around its perimeter.

- Lists of
- Private
  - Public



# Usage Scenarios

## Sharing a Picture With a Group of Friends

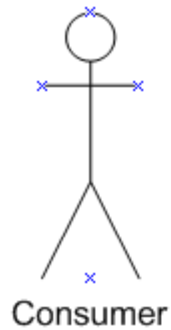


Lists

- None
- Many

# Usage Scenarios

## Sharing a Picture With a Group of Friends

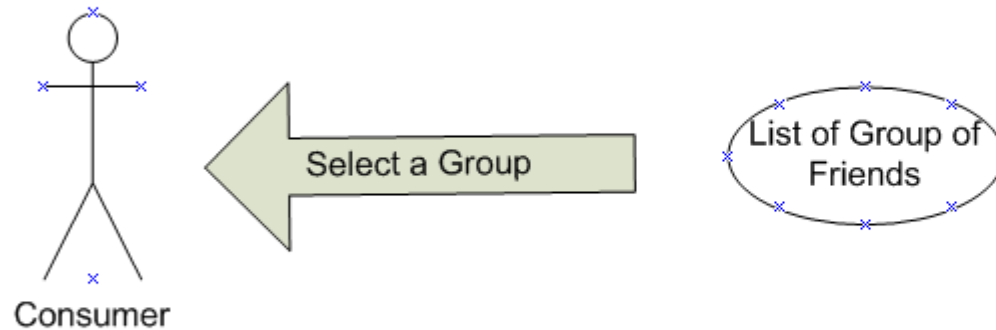


Select by

- Name
- Type
- Preview
- One
- None
- Many

# Usage Scenarios

## Sharing a Picture With a Group of Friends

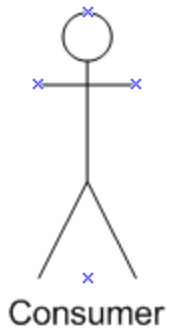


Select by

- Name
- Membership
- Custom made for sharing this picture

# Usage Scenarios

## Sharing a Picture With a Group of Friends



Post Selected Image to Selected Group

Shared Images



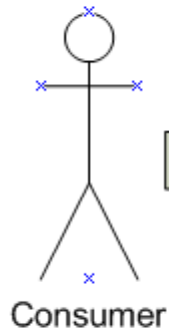
An oval shape representing a group of shared images, with small blue 'x' marks around its perimeter. The text "Shared Images" is written inside the oval.

Posted

- Single image
- Multiple images
- Single group
- Multiple groups

# Usage Scenarios

## Sharing a Picture With a Group of Friends



Tag posted image with some text



### Tagged

- No text
- Some text
- Link
- Different text for different image
- Different text for different groups



# Reality Driven Testing

## Quiz

- Application screens are selected with three controls:

*(a) has 5 options*

*(b) has 6 options*

*(c) has 2 options*

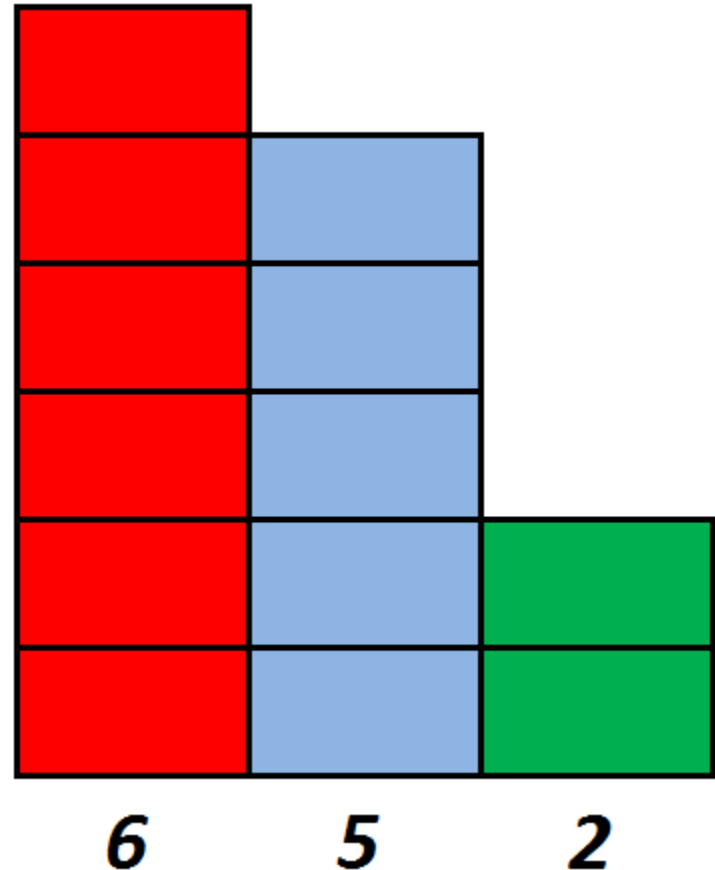
- How many screens can a user choose?



# Reality Driven Testing

## Quiz

- Total Combinations  
=  $6 \times 5 \times 2 = 60$
- To exercise each combination once a total of 60 tests would be required.

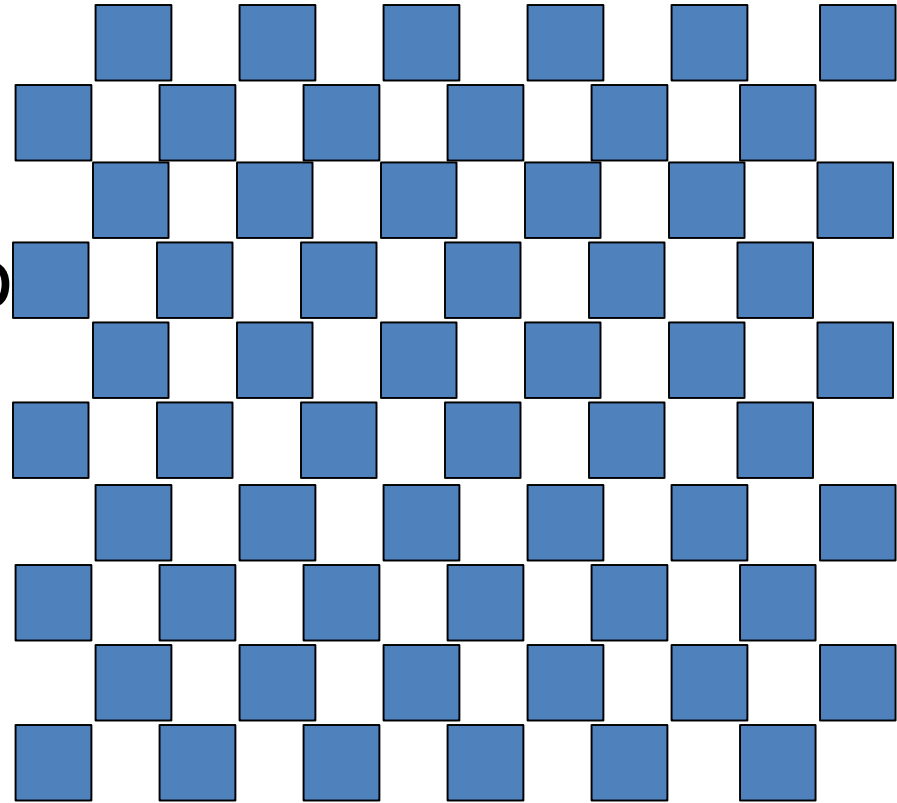




# Reality Driven Testing

## Quiz

- How many tests would be required to exercise all possible screens in every possible order?.



# Reality Driven Testing

## Quiz

- To exercise all screens in every possible order would require

60! Test cases

$$\underline{60! = 60 \times 59 \times 58 \times \dots \times 3 \times 2 \times 1}$$

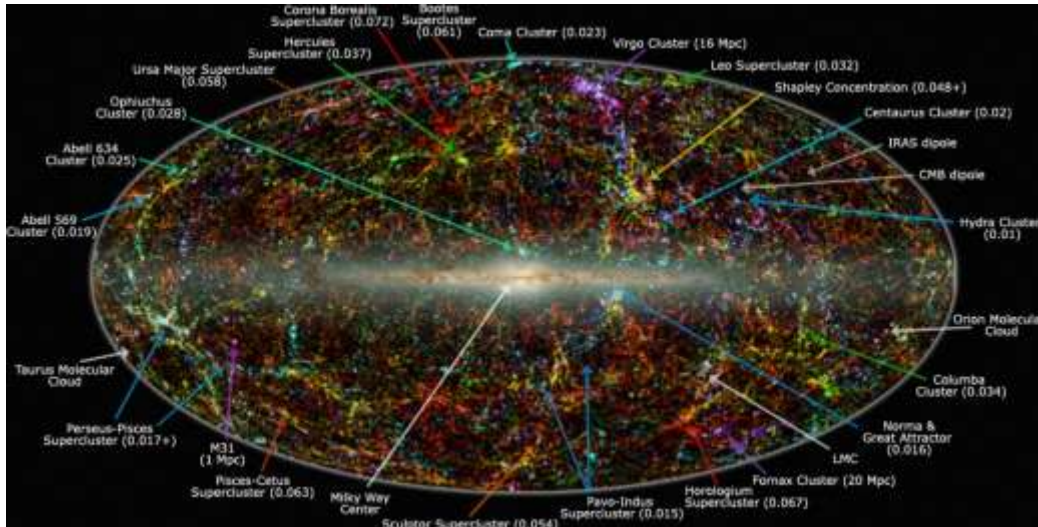
$$\underline{60! \approx 8.32 \times 10^{81}}$$

$$n! > \sqrt{2\pi n} \left(\frac{n}{e}\right)^n .$$

# Reality Driven Testing

## Quiz

How many atoms are in the observable universe?



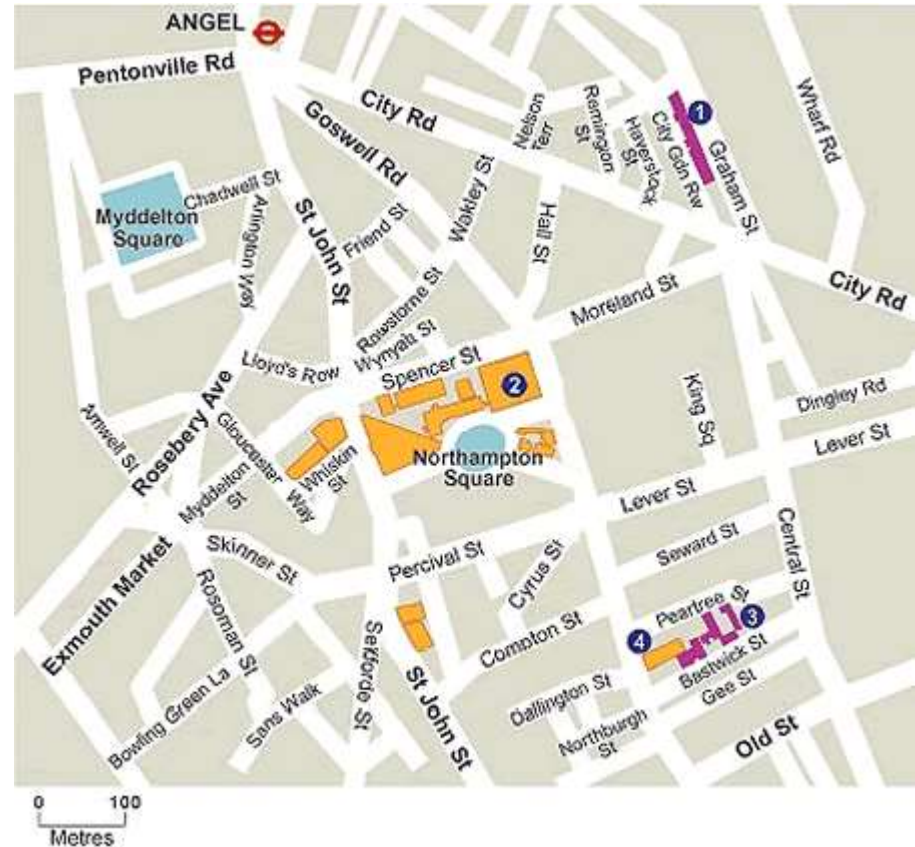
***From  $7.0 \times 10^{79}$***

***To  $1.5 \times 10^{82}$***

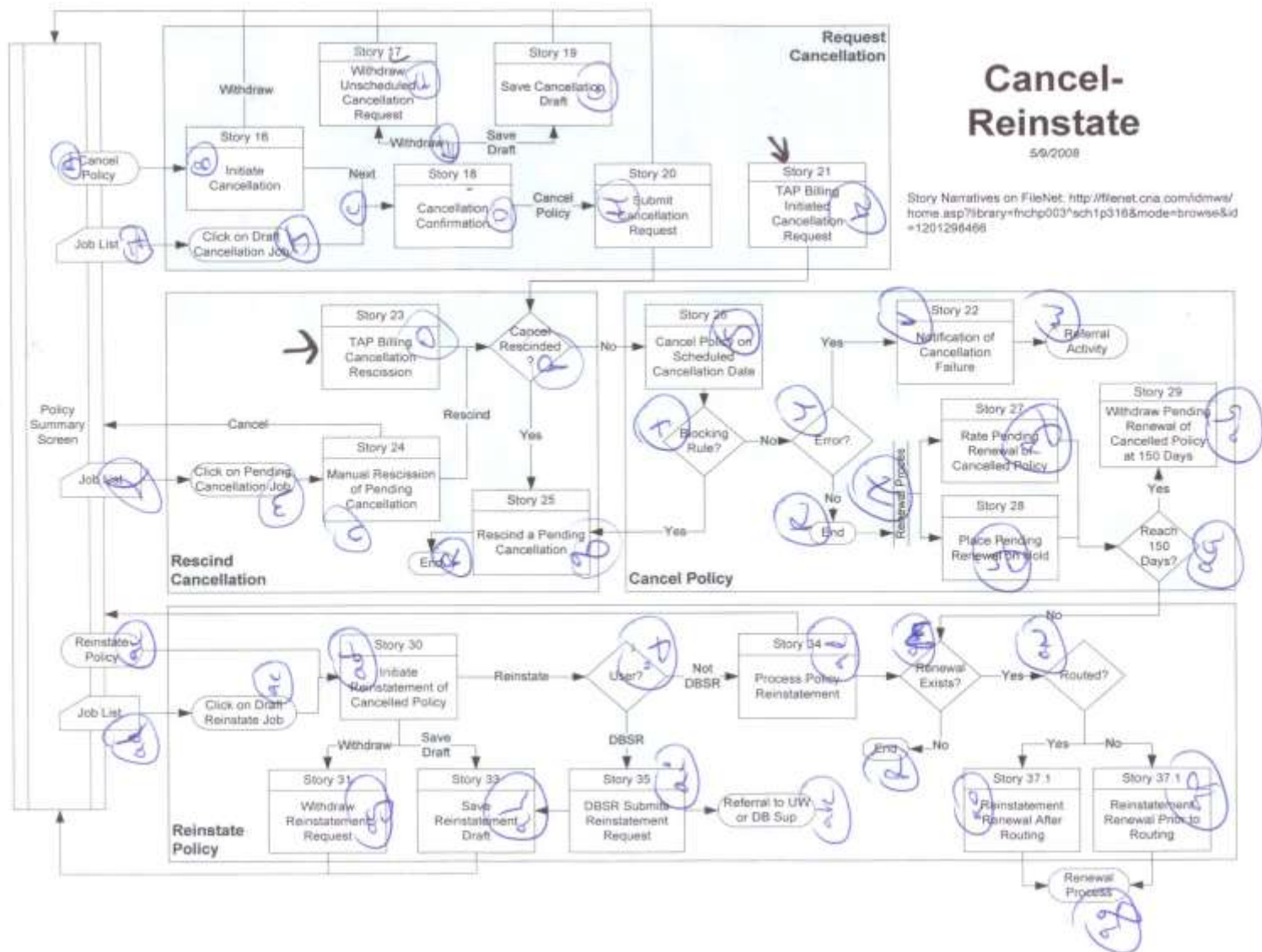
# Charter Types

## Sequences

- Explore paths
- Vary
- Operation order
- Sequences
- Valid
- Invalid
- Multiple
- Concurrent



# Control Flow Testing







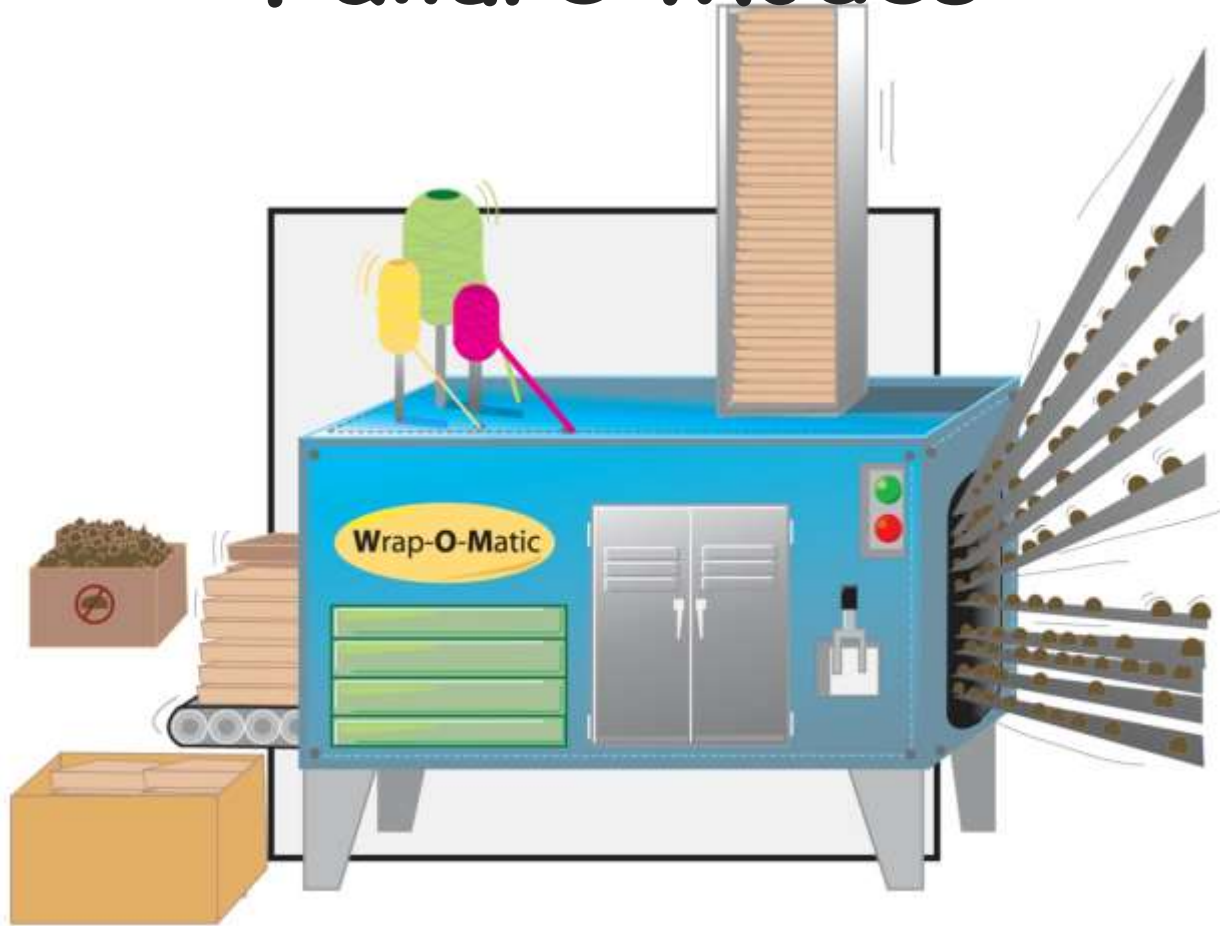




# Charter Types

## Failure Modes

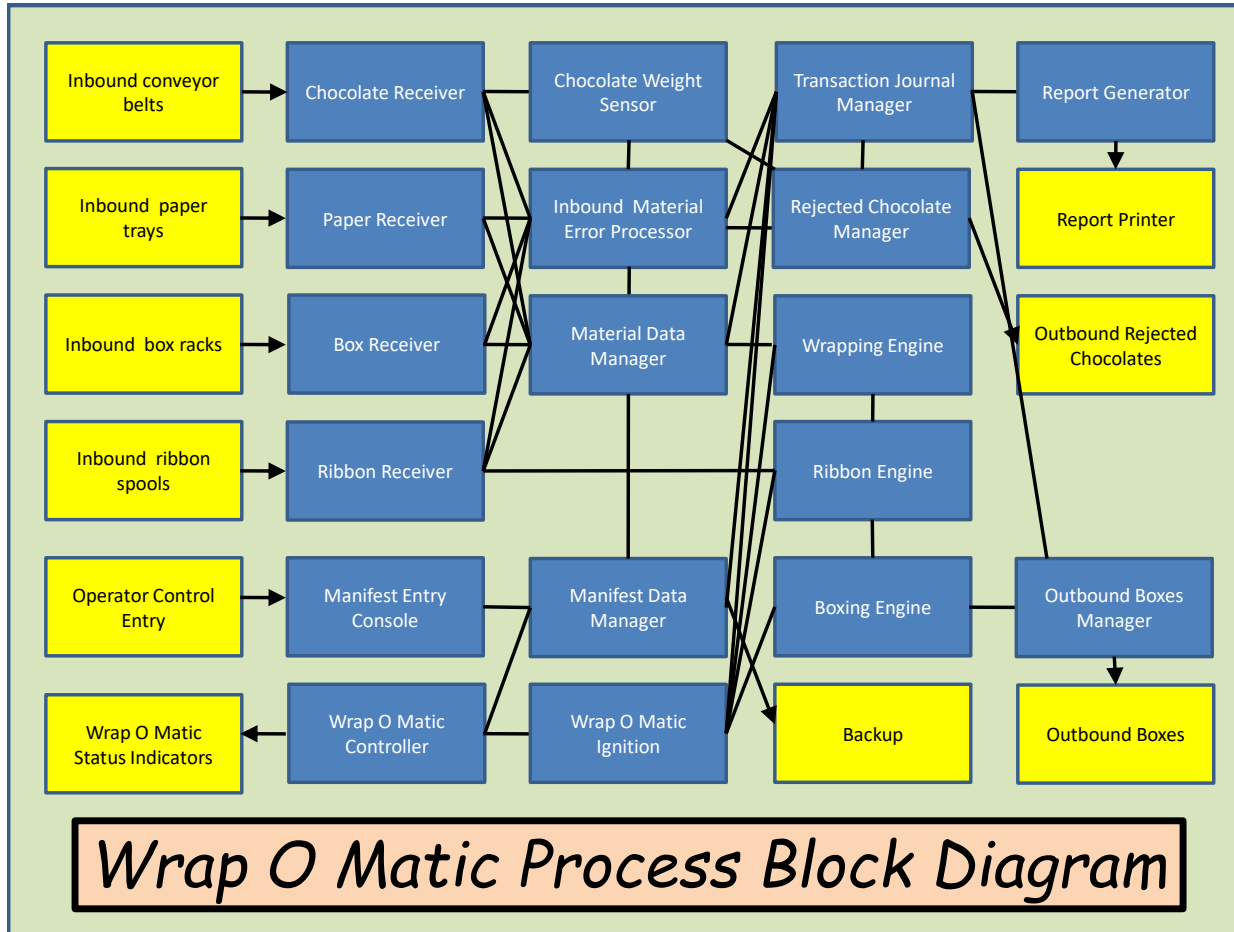
**Wrap-O-Matic**



# Charter Types

## Failure Modes

**Block Diagram**



# Charter Types

## Failure Modes

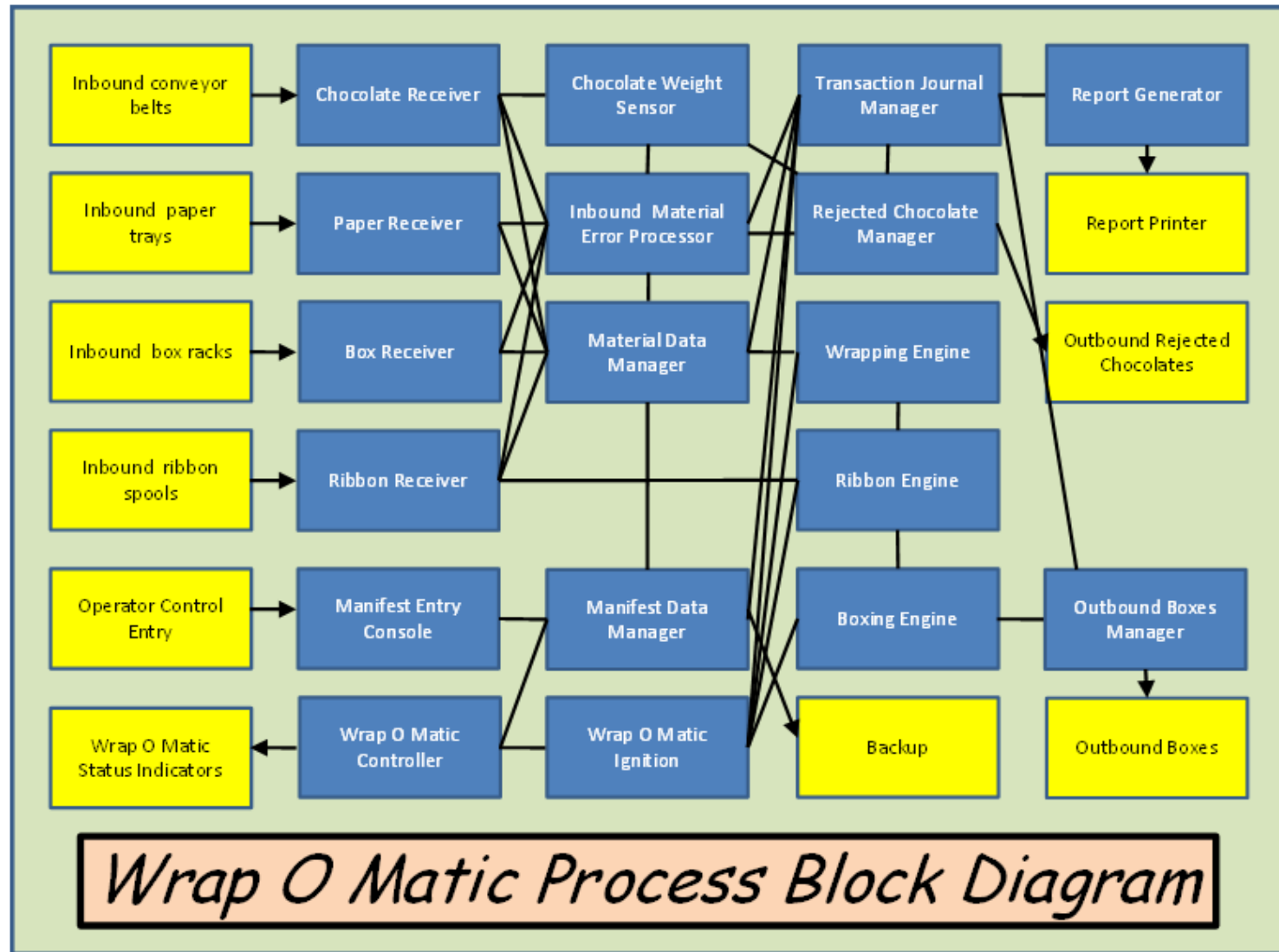
For each object I can ask the question:

What if the object fails during a transaction?

What if the object is not visible?

What if the object is busy?

# Failure Mode Analysis



# Failure Mode Analysis

Wrap O Matic Process Failure Modes		Chocolate Receiver	Paper Receiver	Box Receiver	Ribbon Receiver	Manifest Console	Controller	Weight Sensor	Error Proc	Material Manager	Manifest Manager	Ignition	Journal Manager	Reject Manager	Wrapping Engine	Ribbon Engine	Boxing Engine	Report Generator	Box Manager
User	Scenario	What if the process fails?																	
Operator	Monitor status	S0	S0	S0	S0	S0	S2	S0	S2	S0	S0	S0	S0	S2	S2	S0	S0	S0	S0
Operator	Enter manifests	S0	S0	S0	S0	S2	S2	S0	S0	S3	S2	S0	S0	S0	S0	S0	S0	S0	S0
Operator	Start	S2	S2	S2	S2	S2	S2	S0	S0	S0	S0	S2	S0	S0	S0	S2	S2	S0	S2
Operator	Stop	S0	S0	S0	S0	S0	S2	S0	S0	S0	S0	S2	S0	S0	S0	S0	S0	S0	S0
Operator	Pause	S0	S0	S0	S0	S0	S2	S0	S0	S0	S0	S3	S0	S0	S0	S0	S0	S0	S0
Operator	Resume	S2	S2	S2	S2	S2	S2	S0	S0	S0	S0	S3	S0	S0	S0	S2	S2	S0	S2
Operator	Power Up	S0	S0	S0	S0	S0	S2	S0	S0	S0	S0	S2	S0	S0	S0	S0	S0	S0	S0
Operator	Power Down	S0	S0	S0	S0	S0	S2	S0	S0	S0	S0	S2	S0	S0	S0	S0	S0	S0	S0
Auditor	Batch reports	S0	S0	S0	S0	S2	S2	S0	S0	S2	S2	S0	S2	S3	S2	S0	S0	S2	S0
Auditor	Daily reports	S0	S0	S0	S0	S2	S2	S0	S0	S2	S2	S0	S2	S3	S2	S0	S0	S2	S0
Auditor	Monthly reports	S0	S0	S0	S0	S2	S2	S0	S0	S2	S2	S0	S2	S3	S2	S0	S0	S2	S0
Auditor	Error reports	S0	S0	S0	S0	S2	S2	S0	S2	S2	S2	S0	S2	S2	S2	S0	S0	S2	S0
Loader	Chocolate Wrapping Paper	S0	S2	S0	S0	S0	S0	S0	S0	S2	S0	S0	S0	S0	S0	S0	S0	S0	S0
Loader	Chocolates	S2	S0	S0	S0	S0	S0	S3	S0	S2	S0	S0	S0	S3	S0	S0	S0	S0	S0
Loader	Ribbons	S0	S0	S0	S2	S0	S0	S0	S0	S2	S0	S0	S0	S0	S0	S2	S0	S0	S0
Loader	Empty Boxes	S0	S0	S2	S0	S0	S0	S0	S0	S2	S0	S0	S0	S0	S0	S0	S2	S0	S2
Loader	Box wrapping materials	S0	S0	S0	S0	S0	S0	S0	S0	S2	S0	S0	S0	S0	S0	S0	S0	S0	S0
Unloader	Boxed chocolates	S1	S2	S2	S2	S1	S1	S3	S0	S3	S3	S0	S0	S0	S1	S2	S2	S0	S2
Unloader	Rejected chocolates	S2	S2	S2	S2	S2	S2	S2	S2	S0	S0	S0	S0	S0	S2	S2	S2	S0	S2
Inspector	Contamination	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
Inspector	Ingredient match	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
Inspector	Peanuts	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
Maintainer	Emergency repair	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3
Maintainer	Periodic maintenance	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3
Maintainer	Updates	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3
Maintainer	Upgrades	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3
Maintainer	Cleaning	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3	S3
Maintainer	Configure Backup Devices	S0	S0	S0	S0	S2	S2	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
Maintainer	Schedule Backup	S0	S0	S0	S0	S2	S2	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
Maintainer	Backup	S0	S0	S0	S0	S3	S2	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0
Maintainer	Restore	S0	S0	S0	S0	S2	S2	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0

# Testing Ideas

- Investigative approaches
  - We become truffle snorting pigs and try to find useful information in all evidence we discover
  - We can even get good ideas from out of date sources



# Testing Ideas

Who is paying for the software?

Who is supposed to use the software?

What problem is the software trying to solve?

Does other software have to run before after or during operation?

Will other unrelated software be running at the same time?



# Testing Ideas

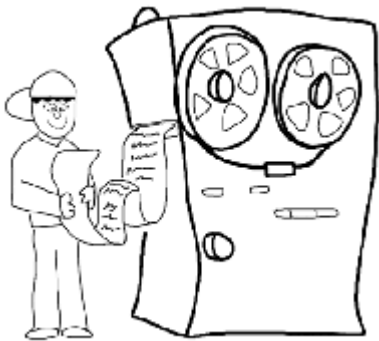
Will data be shared with other software?

Will processing resources be shared with other software?

What are the users going to do with the software?

Are the users familiar with the problem?

Are the users familiar with the solution?





# Testing Ideas

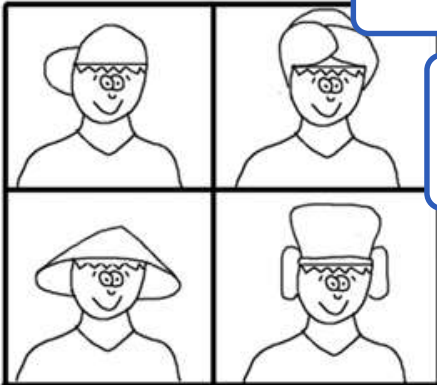
Are there novice users?

Are there expert users?

Are there typical users?

Are there different categories of users doing different tasks?

Who will install the software?



# Testing Ideas

Who will configure the software?

Who will manage the software?

Who will manage the data?

Where will data come from?

Does the software contain elements we do not need?



# Testing Ideas

Are there any non-functional needs?

Does the software have any environmental concerns?

What about operator locale?

Does this software replace existing software?

Does the software replace existing hardware?



# Testing Ideas

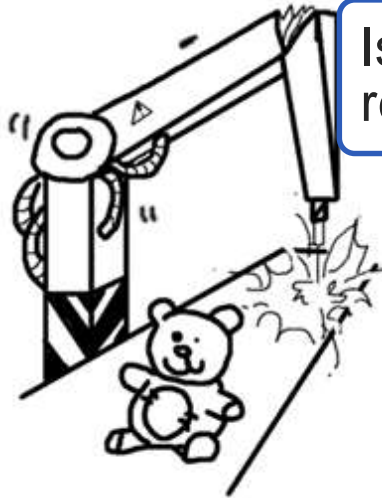
Does the software automate previously manual work?

Is the software being reused?

Is the software being repurposed?

Is the software being customized?

Did the software purchaser understand risk?



# Thank You

- Questions?

