Group Inc. Atlantic
The Project Success Company

Scott Stribrny
President & Managing Director

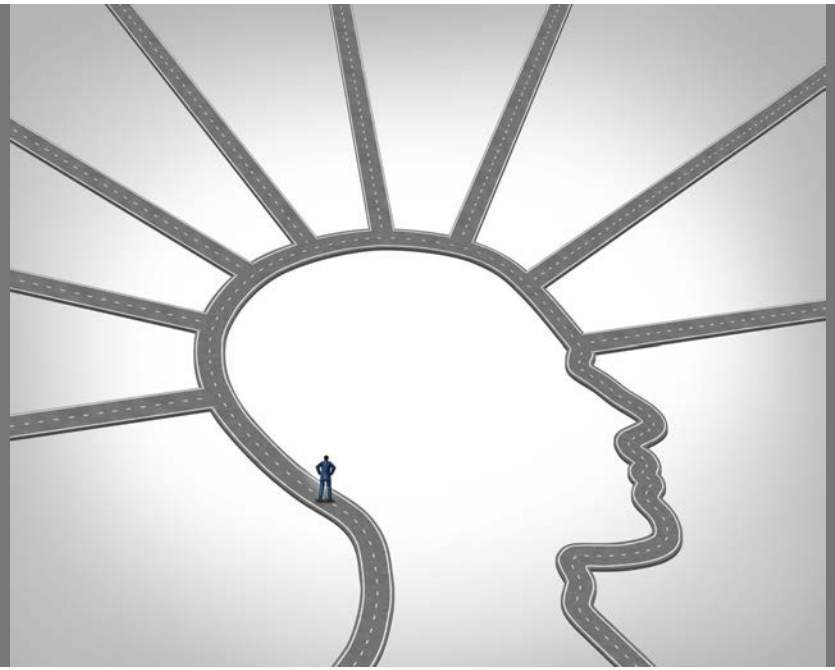
SCAN ME

Risk-Based Testing: Better Focus, Better Products

A Presentation to Milwaukee Software Process Improvement Network

October 20, 2021

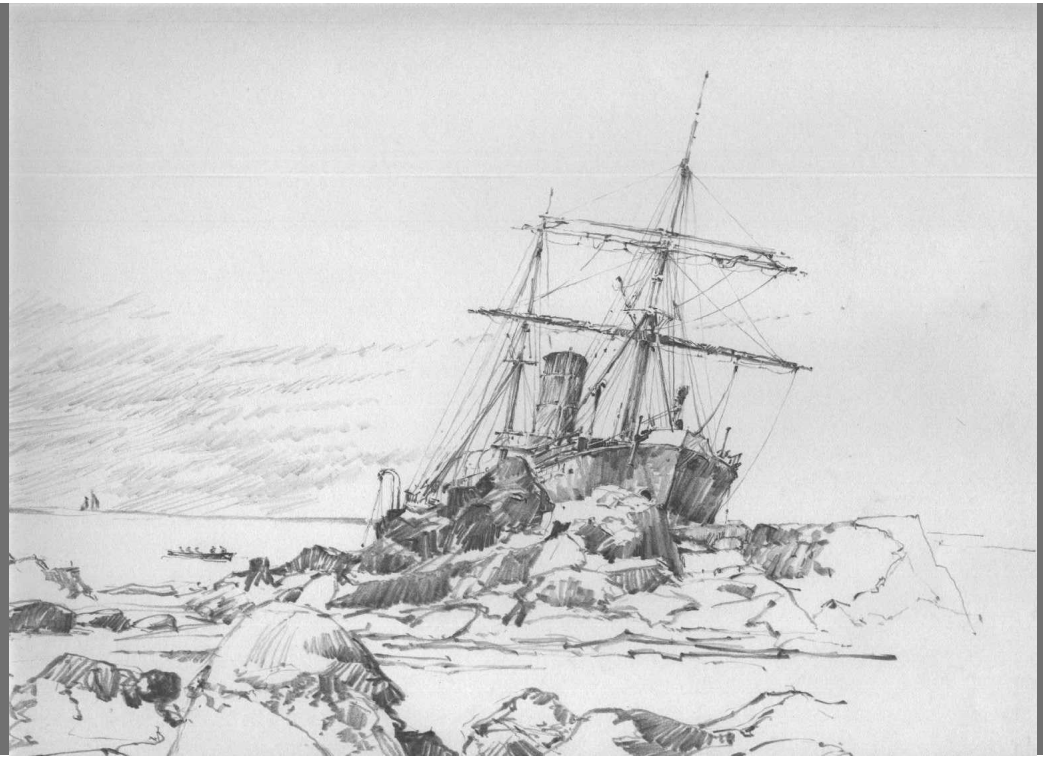
Learning Objectives



- Define Risk-Based testing.
- Explain why risk-based testing is needed.
- Compare 5 techniques for analyzing quality risks.
- Identify a checklist of drivers of quality risk that informs testing strategy.
- Identify a 9-step approach to risk-based testing.

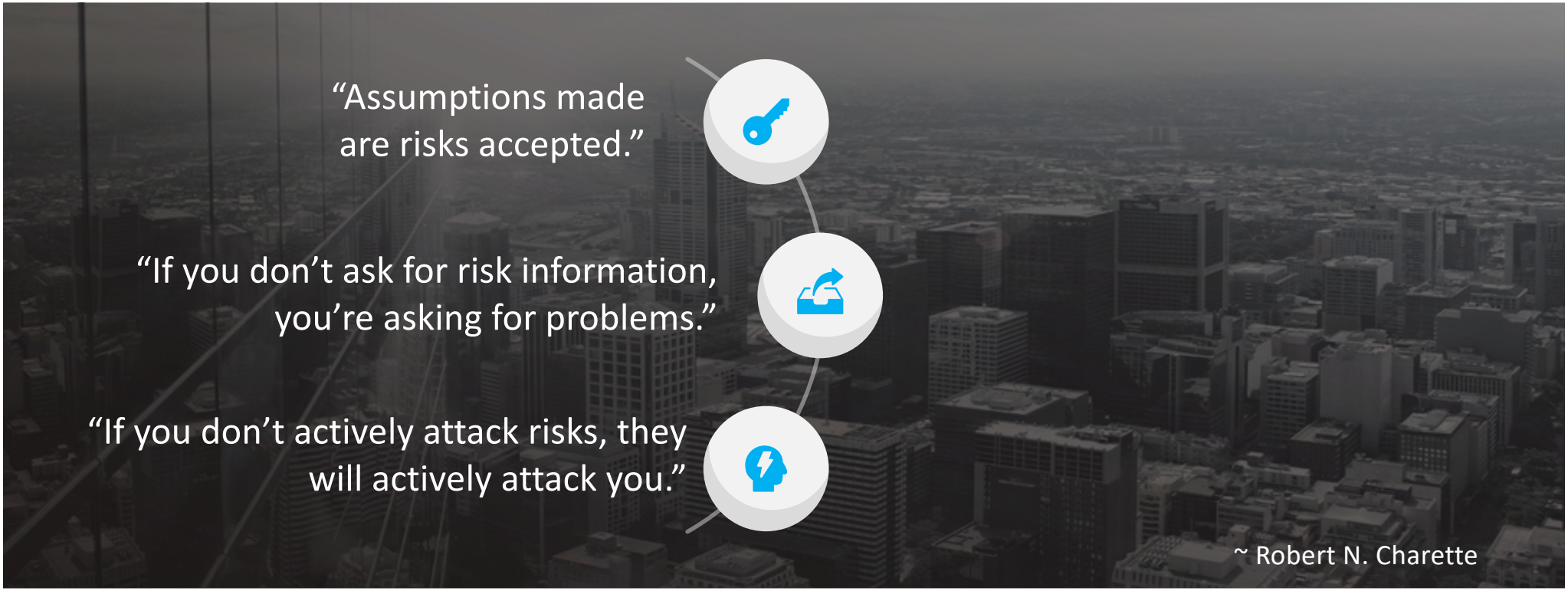
What Are Some of Your Testing Challenges?

What Is Risk Management?



Staying Off the Reef

Risk Management Maxims

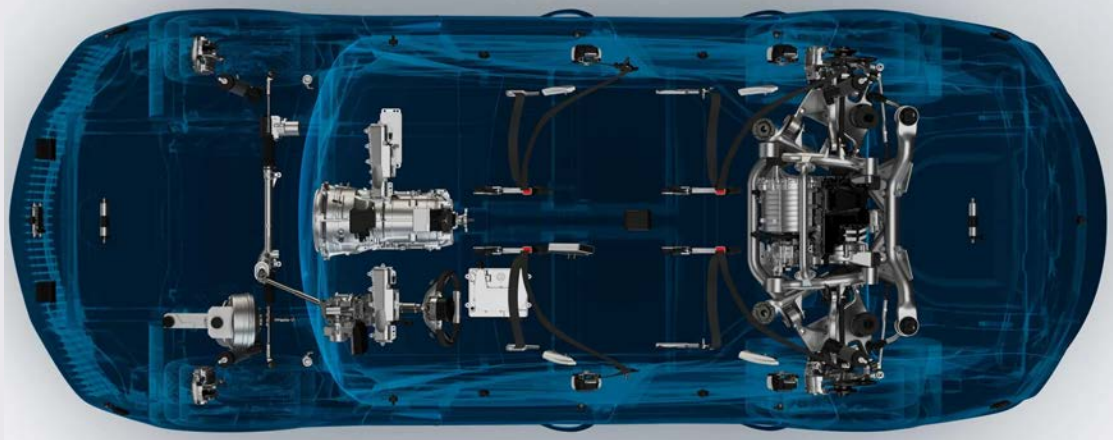


Increasing Size & Importance of Software

How Software Is Eating the Car

The trend toward self-driving and electric vehicles will add hundreds of millions of lines of code to cars. Can the auto industry cope?

By ROBERT N. CHARETTE 07 JUN 2021, IEEE SPECTRUM



Software Was Merely A Part of the Car. … Now Determines Value of the Car.

Testing Real-World Systems

- Testing any real-world system is potentially an infinite task.
- Of this infinite set of possible tests, test managers need to focus on the most significant risks.
- These are the potential failures that are:
  - likely to occur in real-world use
  - would cost a lot if they did occur

When Software Projects Fail

We waste billions of dollars each year on entirely preventable mistakes

By Robert N. Charette

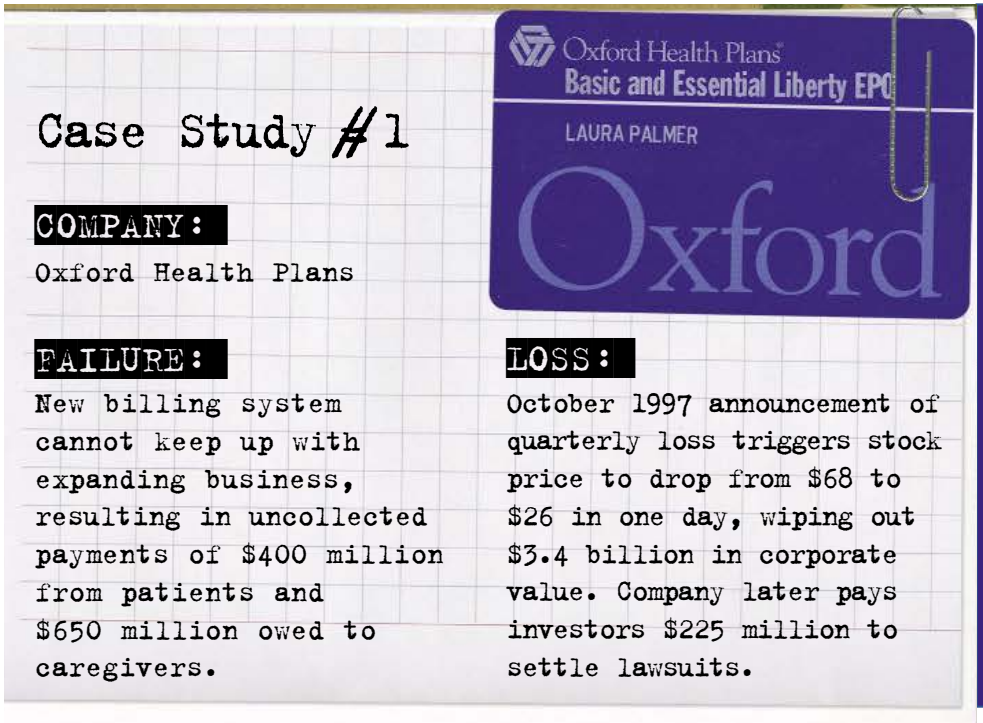
"Why Software Fails," IEEE Spectrum

Software Hall of Shame

YEAR	COMPANY	OUTCOME (COSTS IN US $)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to $33.3 million* loss.
2004–05	UK Inland Revenue	Software errors contribute to $3.45 billion* tax-credit overpayment.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system canceled after $54.5 million† is spent.
2004	Ford Motor Co.	Purchasing system abandoned after deployment costing approximately $400 million.
2004	J Sainsbury PLC [UK]	Supply-chain management system abandoned after deployment costing $527 million.†
2004	Hewlett-Packard Co.	Problems with ERP system contribute to $160 million loss.
2003–04	AT&T Wireless	Customer relations management (CRM) upgrade problems lead to revenue loss of $100 million.
2002	McDonald's Corp.	The Innovate information-purchasing system canceled after $170 million is spent.
2002	Sydney Water Corp. [Australia]	Billing system canceled after $33.2 million† is spent.
2002	CIGNA Corp.	Problems with CRM system contribute to $445 million loss.
2001	Nike Inc.	Problems with supply-chain management system contribute to $100 million loss.
2001	Kmart Corp.	Supply-chain management system canceled after $130 million is spent.
2000	Washington, D.C.	City payroll system abandoned after deployment costing $25 million.
1999	United Way	Administrative processing system canceled after $12 million is spent.
1999	State of Mississippi	Tax system canceled after $11.2 million is spent; state receives $185 million damages.
1999	Hershey Foods Corp.	Problems with ERP system contribute to $151 million loss.
1998	Snap-on Inc.	Problems with order-entry system contribute to revenue loss of $50 million.
1997	U.S. Internal Revenue Service	Tax modernization effort canceled after $4 billion is spent.
1997	State of Washington	Department of Motor Vehicle (DMV) system canceled after $40 million is spent.
1997	Oxford Health Plans Inc.	Billing and claims system problems contribute to quarterly loss; stock plummets, leading to $3.4 billion loss in corporate value.
1996	Arianespace [France]	Software specification and design errors cause $350 million Ariane 5 rocket to explode.
1996	FoxMeyer Drug Co.	$40 million ERP system abandoned after deployment, forcing company into bankruptcy.
1995	Toronto Stock Exchange [Canada]	Electronic trading system canceled after $25.5 million** is spent.
1994	U.S. Federal Aviation Administration	Advanced Automation System canceled after $2.6 billion is spent.
1994	State of California	DMV system canceled after $44 million is spent.
1994	Chemical Bank	Software error causes a total of $15 million to be deducted from 100 000 customer accounts.
1993	London Stock Exchange [UK]	Taurus stock settlement system canceled after $600 million** is spent.
1993	Allstate Insurance Co.	Office automation system abandoned after deployment, costing $130 million.
1993	London Ambulance Service [UK]	Dispatch system canceled in 1990 at $11.25 million**; second attempt abandoned after deployment, costing $15 million.**
1993	Greyhound Lines Inc.	Bus reservation system crashes repeatedly upon introduction, contributing to revenue loss of $61 million.
1992	Budget Rent-A-Car, Hilton Hotels, Marriott International, and AMR [American Airlines]	Travel reservation system canceled after $165 million is spent.

Sources: *Business Week*, *CEO Magazine*, *Computerworld*, *InfoWeek*, *Fortune*, *The New York Times*, *Time*, and *The Wall Street Journal*
* Converted to U.S. dollars using current exchange rates as of press time.
† Converted to U.S. dollars using exchange rates for the year cited, according to the International Trade Administration, U.S. Department of Commerce.
** Converted to U.S. dollars using exchange rates for the year cited, according to the *Statistical Abstract of the United States, 1996*.

Why Is Risk-Based Testing Needed?



- Testing activity may cost up to forty percent of the initial software development value
  - The later a defect is found, the more expensive it is to correct.
  - Test activity commonly does not receive the appropriate attention, because of restrictions of time, resources and cost.
- It's fundamental to find a way to systematically prioritize efforts & allocate resources to the software components that need to be tested carefully.

Risk-Based Testing Responds to Risks

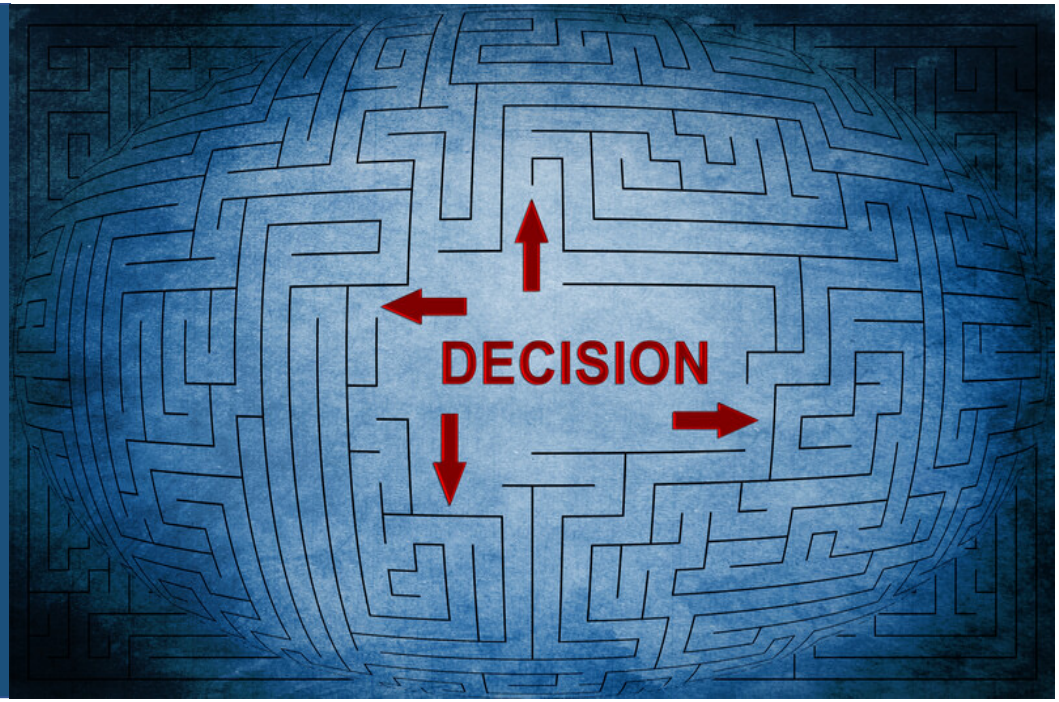
Prioritized Testing: prioritize higher risks and testing them earlier

Targeted Testing: allocating test effort, selecting test techniques and retesting fixes in a way that is appropriate for the level of risk associated with each significant, identified risk

Risk-Based Reporting: reporting test results and status in terms of residual risk, e.g., based on tests which have not yet been run or have been skipped, or defects that have not yet been fixed or retested

Risk-Based Testing

- ✓ Supports making decisions under pressure
- ✓ Contributes to finding the most important defects
- ✓ Makes testing more efficient
- ✓ Supports communication with stakeholders
- ✓ Provides basis for test monitoring & control
- ✓ Provides a basis for bottom-up process improvement



Comparing 5 Techniques for Analyzing Quality Risks

Technique	Informal	ISO 9126/25010	Failure Mode Effect	Hazard	ISO 16085
In a nutshell	Rely on history, experience & checklists	Follow industry-standard quality characteristics	Identify the potential defects & the effects on stakeholders	Analyze causes of hazards (sources of risk)	Follow industry-standard software lifecycle risk management process
Strengths	Easy, light-weight, flexible	Predefined, thorough, common	Precise, meticulous, general	Exact, cautious, systematic	Context flexible, rigorous, proven schedule & cost performance
Weaknesses	Participant-dependent, imprecise	Potentially over-broad, over-regimented	Lengthy, document-heavy, effort-to-learn	Easily overwhelmed by complexity	Requires culture shift
Consider on ____ projects	Low-risk or agile	Standards-compliant	High-risk or conservative	Medical or Safety-critical	Mission critical, strategic
Avoid on ____ projects	Safety-critical or regulated	Very unusual or structure-intolerant	Chaotic, fast-changing, or prototyping	Unpredictable or complex	Minor, small team

Informal

- So, test Function D before/instead of C?
- ‘Assumption Made Is A Risk Accepted?’
- Why?

Use Case	Likelihood	Number of Tests
Function E	0.53	441
Function D	0.15	125
Function C	0.14	117
Function B	0.08	67
Function A	0.06	50
Function G	0.02	17
Function F	0.02	16
Total	**1.00**	**833**

Sources for Likelihood:

- Marketing Analysis of potential customer population/personas
- Usage statistics from previous versions of the product
- User provided estimates

ISO 9126/25010 Software Quality Standard

- Supports a focused, structured, repeatable approach.
- Reduces the likelihood of missing some major risk elements.
  - Six main quality risk categories
  - Two or more sub-characteristics

Failure Mode & Effects Analysis

1. **Imagine** the ways the product could fail.
2. **Ask** questions for each failure mode:
   - ❑ What would that failure look like?
   - ❑ How would you detect that failure?
   - ❑ How expensive would it be to search for that failure?
   - ❑ Who would be impacted by that failure?
   - ❑ How much variation would there be in the effect of the failure?
   - ❑ How serious (on average) would that failure be?
   - ❑ How expensive would it be to fix the underlying cause?
3. **Decide** whether it is cost effective to search for this potential failure.

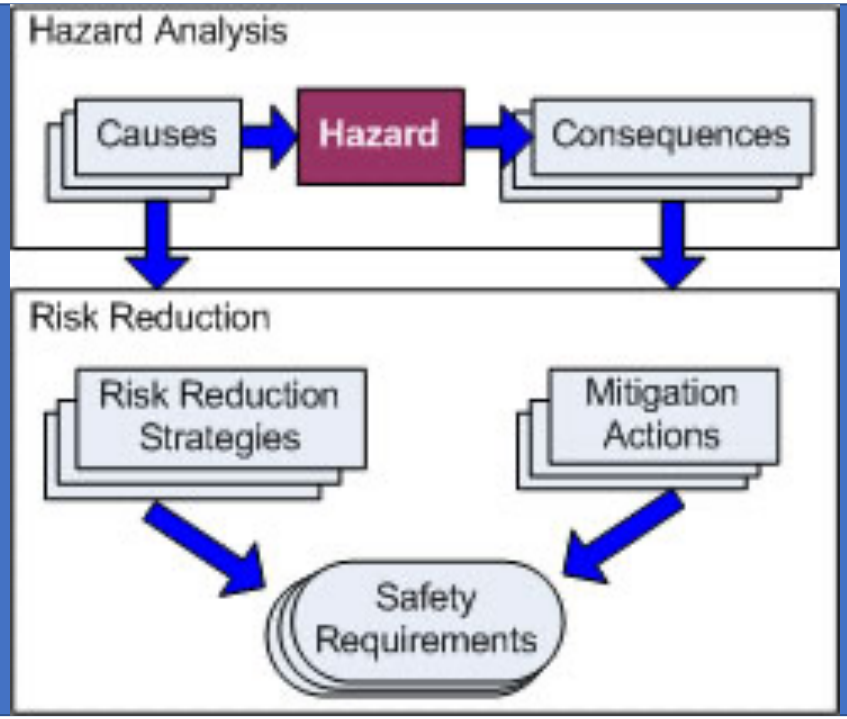
Failure Mode & Effects Analysis (Example)

1. Review the process. ...
2. Brainstorm potential failure modes. ...
3. List potential effects of each failure. ...
4. Assign Severity (SEV) rankings. ...
5. Define Potential Causes. …
6. Assign Occurrence (OCC) rankings. ...
7. Identify Current Process controls. …
8. Assign Detection (DET) Method & Rankings. ...
9. Calculate the Risk Priority Number (RPN). …
10. Recommend Risk Aversion Action for High RPN.

Process Step	Potential Failure Mode	Potential Failure Effect	SEV[1]	Potential Causes	OCC[2]	Current Process Controls	DET[3]	RPN[4]	Action Recommended
What is the step?	In what ways can the step go wrong?	What is the impact on the customer if the failure mode is not prevented or corrected?	How severe is the effect on the customer?	What causes the step to go wrong (i.e., how could the failure mode occur)?	How frequently is the cause likely to occur?	What are the existing controls that either prevent the failure mode from occurring or detect it should it occur?	How probable is detection of the failure mode or its cause?	Risk priority number calculated as SEV x OCC x DET	What are the actions for reducing the occurrence of the cause or for improving its detection? Provide actions on all high RPNs and on severity ratings of 9 or 10.
ATM Pin Authentication	Unauthorized access	• Unauthorized cash withdrawal • Very dissatisfied customer	8	Lost or stolen ATM card	3	Block ATM card after three failed authentication attempts	3	72	
	Authentication failure	Annoyed customer	3	Network failure	5	Install load balancer to distribute work-load across network links	5	75	
Dispense Cash	Cash not disbursed	Dissatisfied customer	7	ATM out of cash	7	Internal alert of low cash in ATM	4	196	Increase minimum cash threshold limit of heavily used ATMs to prevent out-of-cash instances
	Account debited but no cash disbursed	Very dissatisfied customer	8	• Transaction failure • Network issue	3	Install load balancer to distribute work-load across network links	4	96	
	Extra cash dispensed	Bank loses money	8	• Bills stuck to each other • Bills stacked incorrectly	2	Verification while loading cash in ATM	3	48	

1. **Severity**: Severity of impact of failure event. It is scored on a scale of 1 to 10. A high score is assigned to high-impact events while a low score is assigned to low-impact events.
2. **Occurrence**: Frequency of occurrence of failure event. It is scored on a scale of 1 to 10. A high score is assigned to frequently occurring events while events with low occurrence are assigned a low score.
3. **Detection**: Ability of process control to detect the occurrence of failure events. It is scored on a scale of 1 to 10. A failure event that can be easily detected by the process control is assigned a low score while a high score is assigned to an inconspicuous event.
4. **Risk priority number**: The overall risk score of an event. It is calculated by multiplying the scores for severity, occurrence and detection. An event with a high RPN demands immediate attention while events with lower RPNs are less risky.

Hazard Analysis (Objectives)

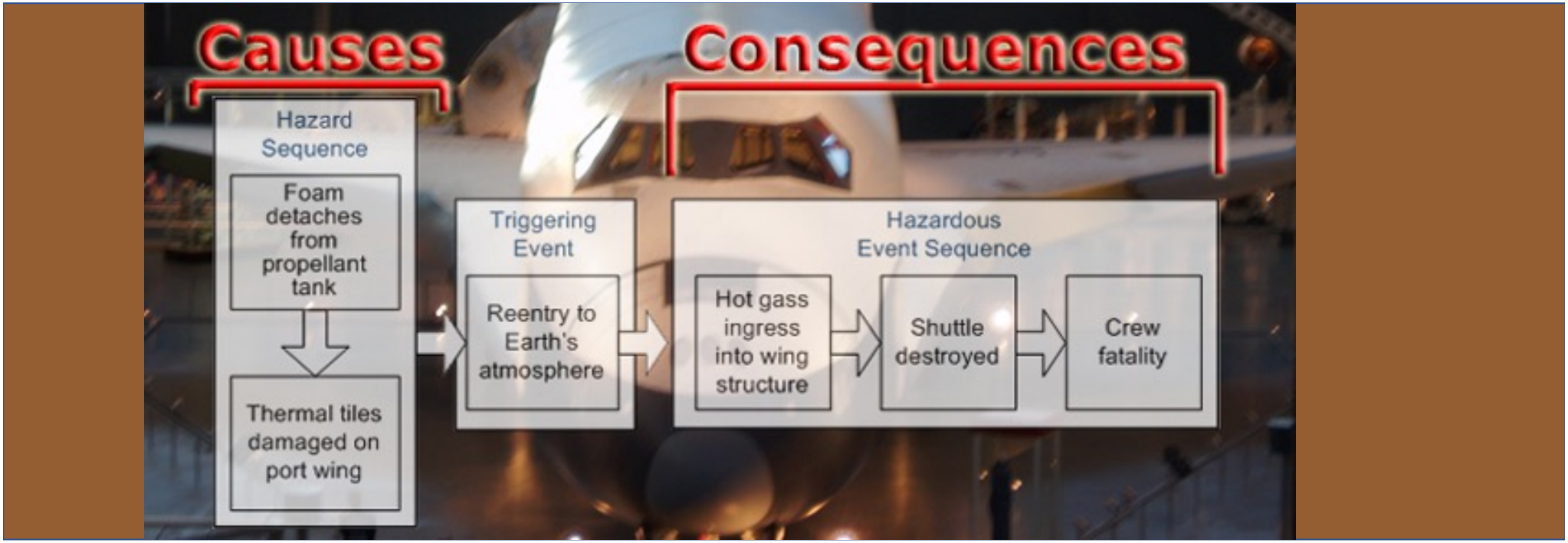


Identify hazards. To determine the hazards and hazardous events of the equipment under control and the control system (in all modes of operation), for all reasonably foreseeable circumstances including fault conditions and misuse.

Identify causes. To analyze the event sequences leading to the hazardous events identified

Determine risks. To analyze the risks associated with the hazardous events.

Hazard Analysis (example)



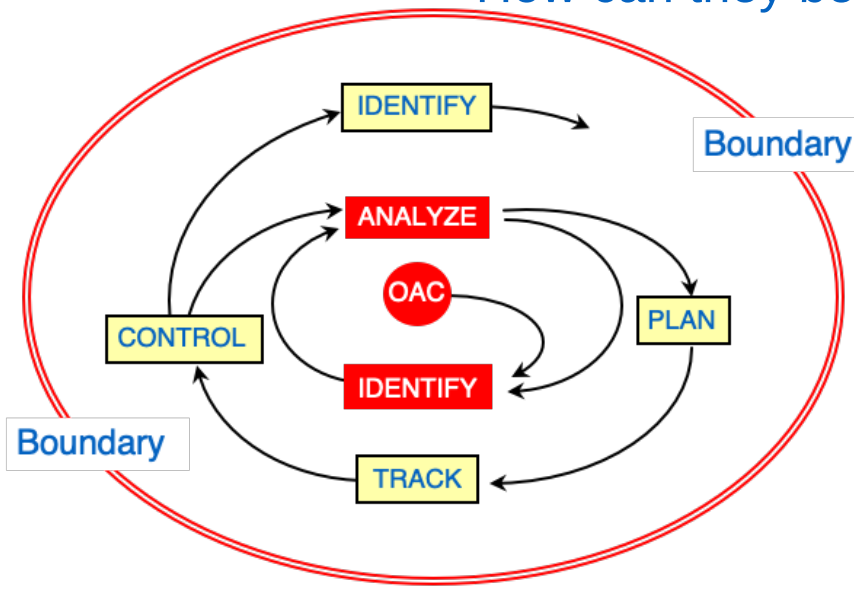
What Managers Want to Know About Risks

1. What are today's risks - are they higher or lower than before?
2. Are the risks likely to get higher or lower in the future?
3. What is being done to reduce risks, to monitor risks and to prevent risks in the future?
4. Who is responsible for the aversion measures - who can I call if things are not correct?
5. How will I know the aversion measures are being put into place?
6. What is the timetable for the aversion measures?
7. How and what should I communicate concerning risk internally, to suppliers, and to the customer?

Based on the principles of Dr. Robert Charette

16085–Proven Process Answers Management Questions

- What are the **Objectives**, **Assumptions** & **Constraints**?
- What are the risks?
- How can they be categorized?



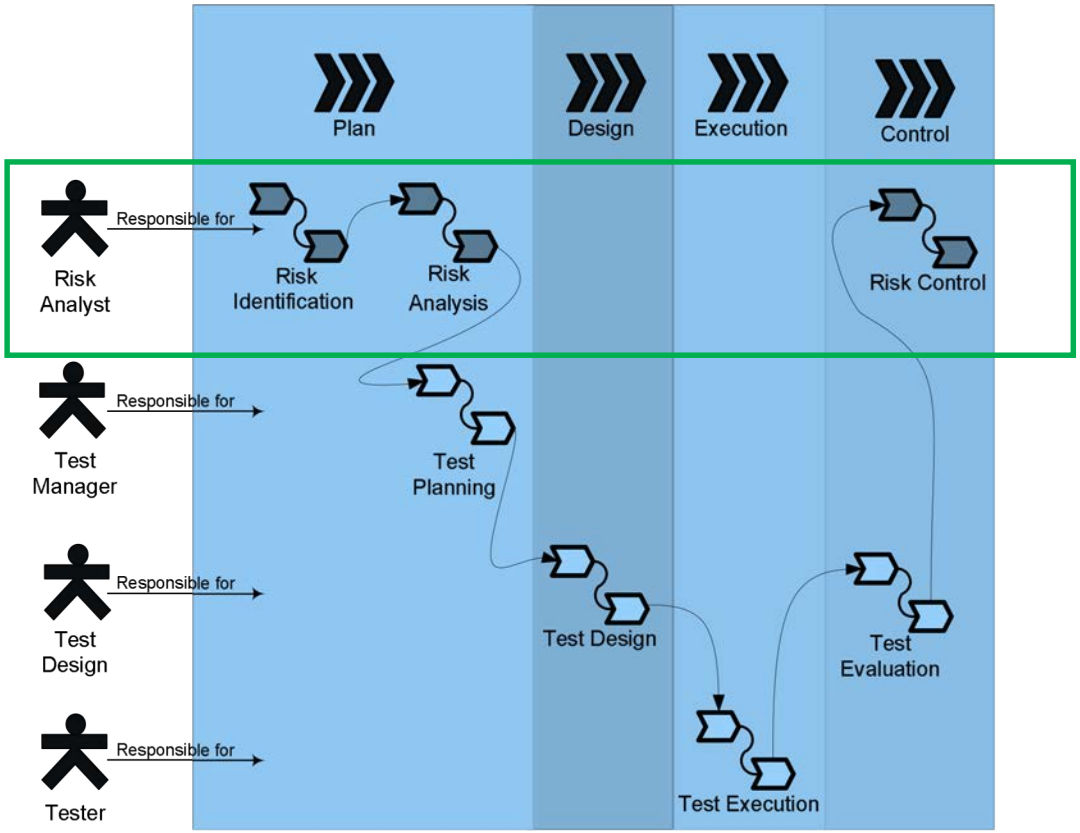
- What is the risk's likelihood of occurring?
- What is the consequence of the risk?
- What is the timing?

- What is considered an acceptable risk?
- What is the total exposure to risk?
- Is the risk acceptable?
- What is the priority?
- What other choices exist to avert the risk?

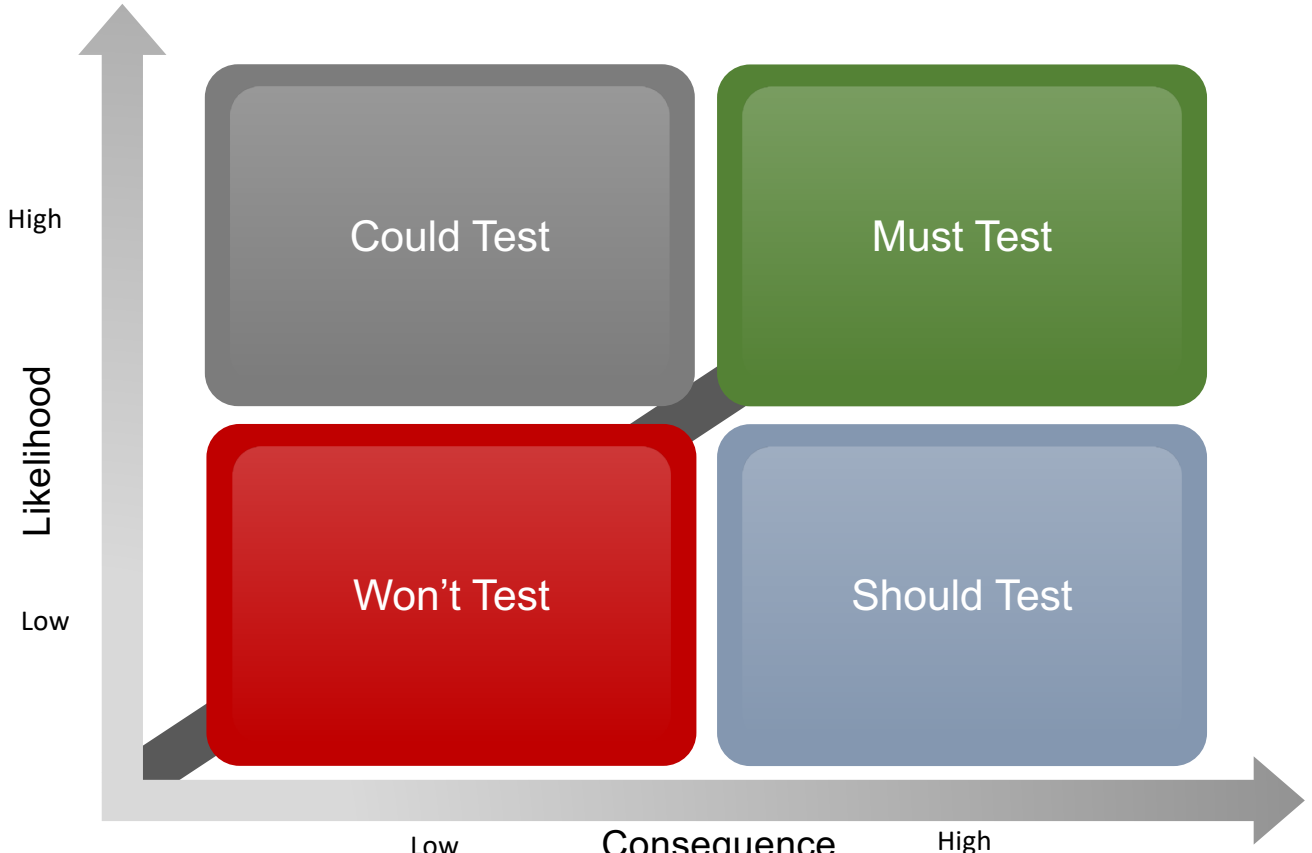
ISO/IEC/IEEE 16085 Systems and software engineering — Life cycle processes — Risk management

~ Chaired by Dr. Robert N. Charette

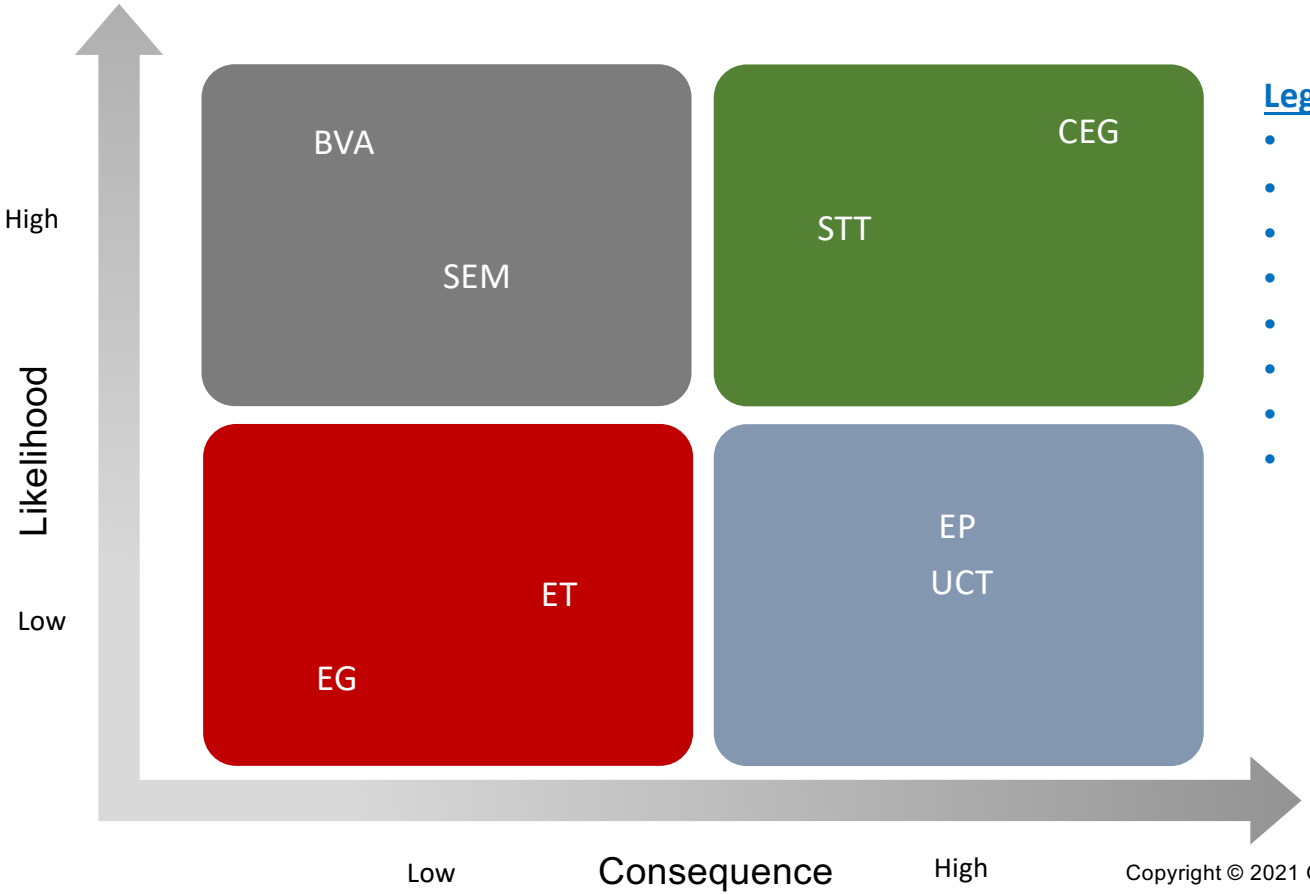
Risk-Based Testing Process



Prioritization of Testing Based on Level of Risk Exposure



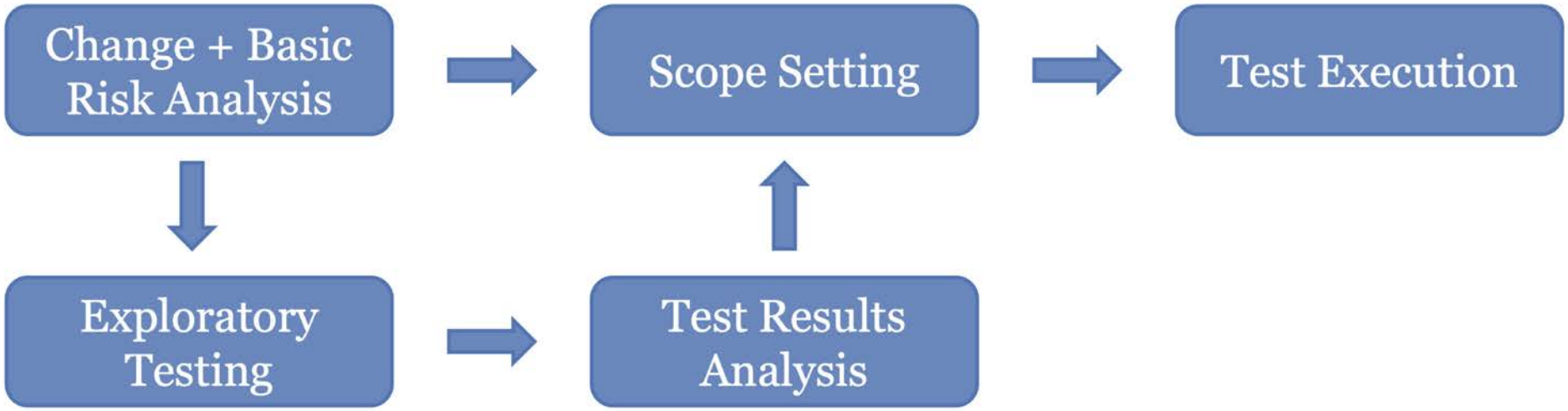
Technique Emphasis Based on Level of Risk Exposure



Legend

- BVA: Boundary Value Analysis
- CEG: Cause/Effect Graphing
- EG: Error Guessing
- EP: Equivalence Partitioning
- ET: Exploratory Test
- RLT: Real Life Test
- STT: State Transition Testing
- UCT: Use Case Testing

Analysis Informed by Exploratory Testing



"To the extent that the next test we do is influenced by the result of the last test we did, we are doing exploratory testing."

~ James Bach

Extent of Testing Based on Risk Exposure

Risk Exposure	Extent of Testing	Comments
Very low	None	**Only report** defects observed in these risk areas.
Low	Opportunistic	Leverage other tests or activities to run a test or two of an interesting condition in the related risk area, but only if it involves a very **small investment of time and effort** and only if the opportunity presents itself.
Medium	Cursory	Run a **small number of tests** that sample the most interesting conditions in the related risk areas.
High	Broad	Run a **medium number of tests** that exercise many different interesting conditions in the related risk areas.
Very high	Extensive	Run a **large number of tests** that are both broad and deep, where deep tests exercise many combinations and variations of interesting conditions.

Risk Exposure (Look Up Table)

Consequence

Likelihood	Very High	High	Medium	Low	Very Low
Very High	Very High	Very High	High	High	Medium
High	Very High	High	High	Medium	Medium
Medium	High	High	Medium	Low	Low
Low	High	Medium	Low	Low	Very Low
Very Low	Medium	Medium	Low	Very Low	Very Low

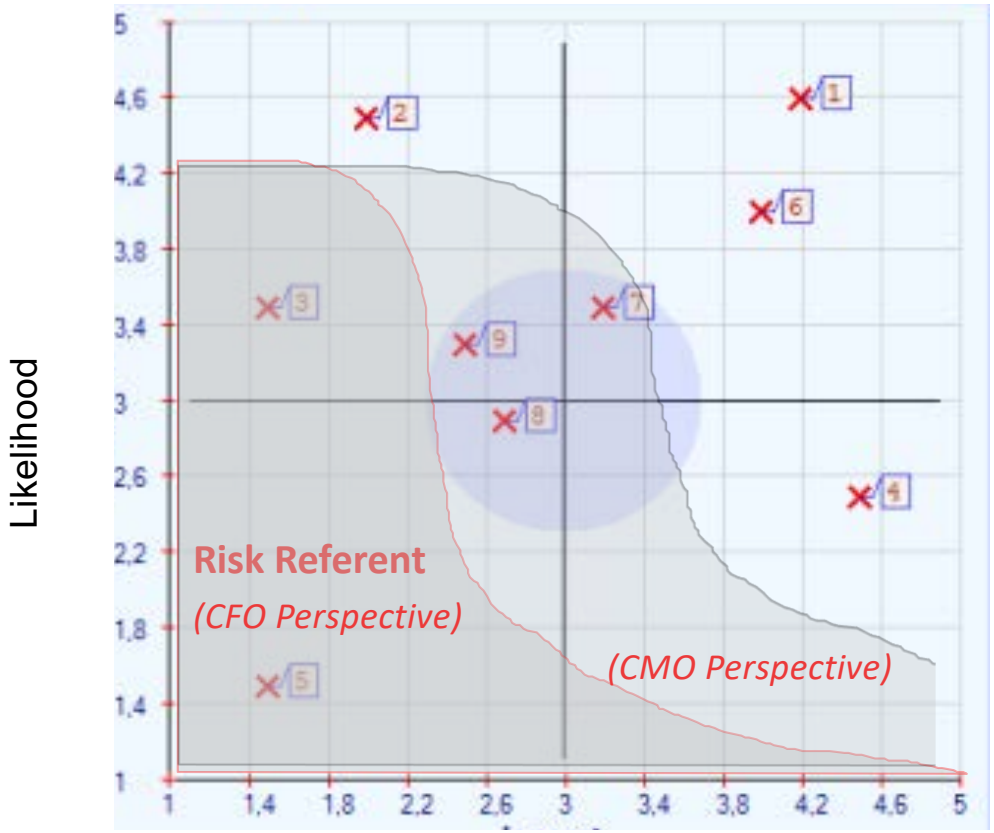
Weighted Scoring Based on Risk Categories

	Likelihood		Consequence			Exposure
Area to Test	**Usage Frequency**	**Visibility**	**Functionality**	**Usability**	**Maintainability**	
Weight	**3**	**10**	**3**	**1**	**3**	
Item A	5	3	2	4	5	**1125**
Item B	5	3	5	4	5	**1530**
Item C	2	1	2	2	5	**368**
Item D	1	1	4	2	5	**377**
Item E	4	4	3	2	0	**572**
Item F	5	0	4	1	1	**240**

Prioritization of Testing

Likelihood Drivers:

- **Size**
- **Complexity**
- **New Development**
- **Level of Re-uses**



Consequence Drivers:

- **Safety**
- **Competitive Differentiation**
- **Financial Loss**
- **Brand Erosion**

Let’s Brainstorm Drivers of Quality Risk

Drivers of Quality Risks Checklist–1

- **New things**: less likely to have revealed its bugs yet.
- **New technology:** same as new code, plus the risks of unanticipated problems.
- **Learning curve:** people make more mistakes while learning.
- **Changed things**: same as new things, but changes can also break old code.
- **Poor control**: without SCM, files can be overridden or lost.
- **Late change:** rushed decisions, rushed or demoralized staff lead to mistakes.
- **Rushed work:** some projects are under-funded and all aspects of work quality suffer.

Adapted from "Black BoxTesting", by Cem Kaner & James Bach

Drivers of Quality Risks Checklist–2

- **Fatigue:** tired people make mistakes.
- **Distributed team:** a far flung team communicates less.
- **Other staff issues:** programmers who won't talk to each other (neither will their code)…
- **Surprise features:** features not carefully planned may have unanticipated effects on other features.
- **Third-party code:** external components may be much less well understood than local code, and much harder to get fixed.
- **Ambiguous**: ambiguous descriptions (in specs or other docs) lead to incorrect or conflicting implementations.

Adapted from "Black BoxTesting", by Cem Kaner & James Bach

Drivers of Quality Risks Checklist–3

- **Conflicting requirements:** ambiguity often hides conflict, result is loss of value for some person.
- **Mysterious silence**: when something interesting or important is not described or documented, it may have not been thought through, or the designer may be hiding its problems.
- **Unknown requirements**: requirements surface throughout development. Failure to meet a legitimate requirement is a failure of quality for that stakeholder.
- **Evolving requirements**: people realize what they want as the product develops. Adhering to a start-of-the-project requirements list may meet the contract but yield a failed product.

Adapted from "Black BoxTesting", by Cem Kaner & James Bach

Drivers of Quality Risks Checklist–4

- **Defect Density:** anything known to have lots of problems has more.
- **Recent failure**: anything with a recent history of problems.
- **Upstream dependency**: may cause problems in the rest of the system.
- **Downstream dependency**: sensitive to problems in the rest of the system.
- **Distributed**: anything spread out in time or space, that must work as a unit.
- **Complex**: what's hard to understand is hard to get right.

Adapted from "Black BoxTesting", by Cem Kaner & James Bach

Test Execution Status

	Risk Level	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Risk Item 1	Very High	✓			✓	✓	✓		
Risk Item 2	Very High	X			?	X	✓		
Risk Item 3	High		✓		?	?		✓	
Risk Item 4	Medium		X	✓					
Risk Item 5	Low								?

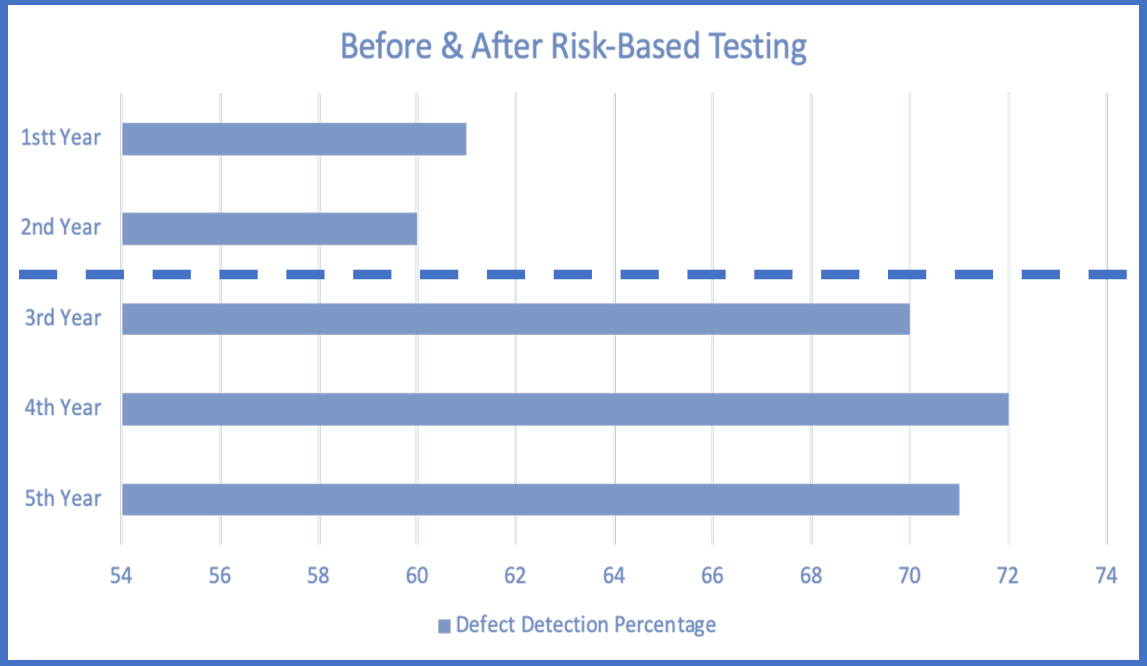
Legend:
✓: test executed and passed
X: test executed and failed
?: test has not yet been executed

9-Step Approach to Risk-Based Testing

1. Get risk ideas from risk factors or from failure mode categories (aka risk taxonomies).
2. For each important idea, determine test activities, prepare tests (that have power against that idea), shift resources to gather information about it.
3. Maintain traceability between risks and tests.
4. Monitor and report the status of the risks as the project goes on and you learn more about them.
5. Assess coverage of the testing effort program, given a set of risk-based tests. Find holes in the testing effort.
6. If a risk is determined to be small enough, then stop testing against it.
7. On retesting an area, evaluate your tests experiences so far to determine what risks they were testing for and whether more powerful variants can be created.
8. Do at least some non-risk-based testing, to cover yourself in case your risk analysis is wrong.
9. Build risk taxonomies from lists of defect histories, configuration problems, tech support requests and obvious customer confusions -- deepen your lists of failure modes

Better Focus, Better Products

- Train stakeholders on Risk-Based Testing
- Lead quality risk analysis workshops
- Align tests with quality risks
- Coach adoption of practices
- Assess effectiveness & efficiencies
- Report on quality improvements



DDP = “The number of defects found by a test level, divided by the number found by that test level and any other means afterwards.” (ISTQB Glossary)

Risk-Based Testing (Issues)

- ✓ Unrecognized risks and risks that are assessed to be too low.
  - Only causes problems if the risks will become a reality
  - Emphasizes the importance of rigorous risk identification and analysis processes
- ✓ Risk assessment can be based on too subjective criteria.
  - Lack of reliable objective criteria
  - Quite common to trust to “expert” judgments.
- ✓ Difficulty to identify & select the right stakeholders for risk assessment.
  - E.g., if a customer is asked to participate in risk assessment, it can be quite a surprise for the customer in some cases that the product is not tested fully.
- ✓ Difficulty attaching a test to an identified risk
  - risks are described too abstractly.

Risk-Based Testing **(Benefits)**

- ✓ Informs when to stop testing.
- ✓ Reduces & focuses test cases on the most critical areas.
- ✓ Less & more efficient test cases can be specified.
- ✓ Discovers problem areas early.
- ✓ Adjusts overall test goals, strategies, & directions for testing against priority problems.

Scott Stribrny

SCAN ME