Why Are You NOT Performing Risk Analysis to Improve Your Testing?

A Risk Analysis Approach to Testing

SOFTWARE TEST PROFESSIONALS
SPRING CONFERENCE 2019

1

Speaker Information

Bob Crews
bcrews@checkpointech.com
Office: 813-818-8324
Direct: 813-493-3678

- **CEO/Co-founder**
  Checkpoint Technologies, Inc.
- **President**
  TBQAA (Tampa Bay Quality Assurance Association)
- **Community Director**
  Vivit Board of Directors (Micro Focus' Independent Software User Group)
- **Co-leader**
  Florida Vivit Chapter

2

2

Agenda

What We'll Discuss

- Why you can't test everything
- The value of performing Risk Analysis
- How to improve your testing with Risk Analysis

3

> "You can't test a program completely"

4

Three Reasons Complete Testing is Impossible

1

The domain of possible inputs is *too large* to test

2

Too many possible paths through the program to test

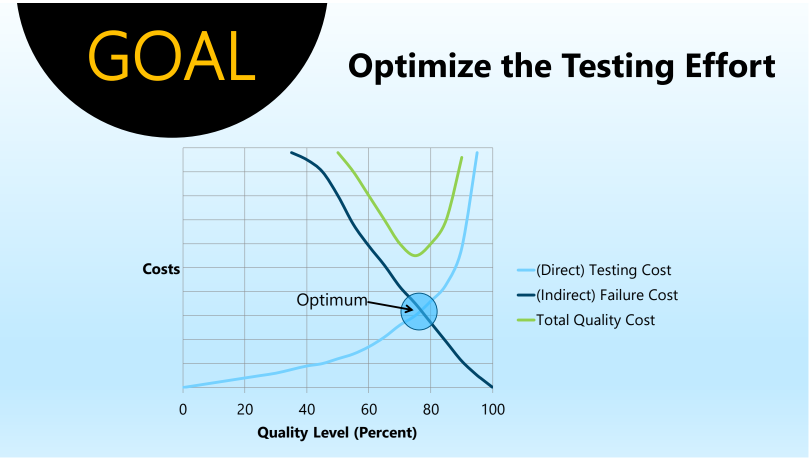
3

User Interface issues are *too complex*

The Point: there are ***numerous*** scenarios and conditions which must be validated

"Testing Computer Software" by Cem Kaner, Jack Falk, and Hung Nguyen

5



GOAL

Optimize the Testing Effort

6

> “Risk comes from not knowing what you’re doing.”
>
> —WARREN BUFFET

7

ASSET

Or object of the protection efforts, can be a system component, data, requirement, test or even a complete system

IMPACT or CRITICALITY

On the organization, were the risk to be realized, can be monetary, reputation, or breach of a law, regulation, or contract

PROBABILITY is the LIKELIHOOD

That a given event will be triggered

EXPOSURE

Represents the number of users impacted and/or the “importance” of the users impacted

8

Risk-Based Testing Approaches

9

A

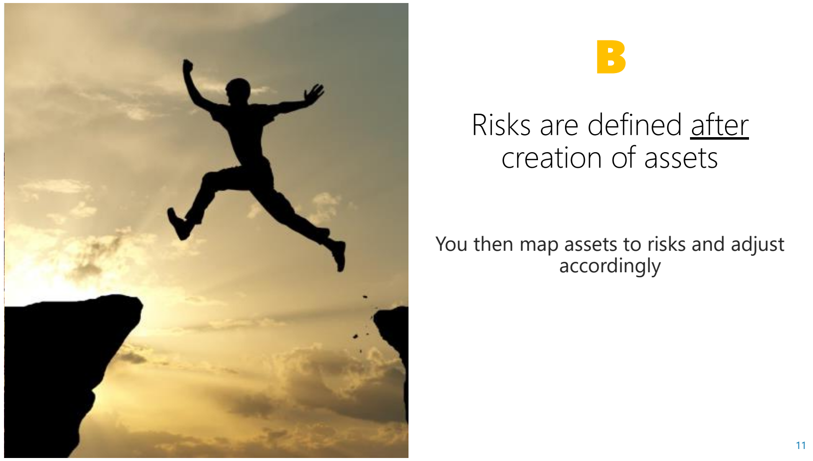
Risks are defined before all else

All assets **evolve** from the risks

This is true risk-based approach

Must start very early

10



B

Risks are defined after creation of assets

You then map assets to risks and adjust accordingly

11

C

Risks are “implied” by assets failing

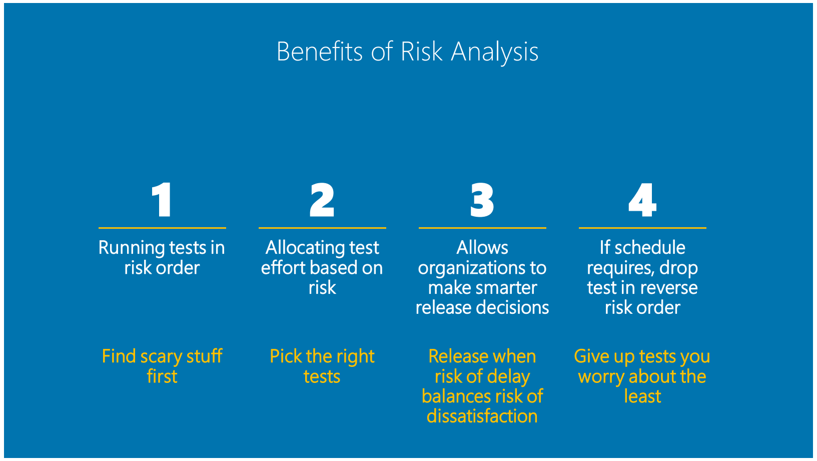
Perform **risk analysis**

Assign risk scores based upon IMPACT and PROBABILITY

Ask “What is the business impact if this fails?”

Ask “What is the probability of this failing?”

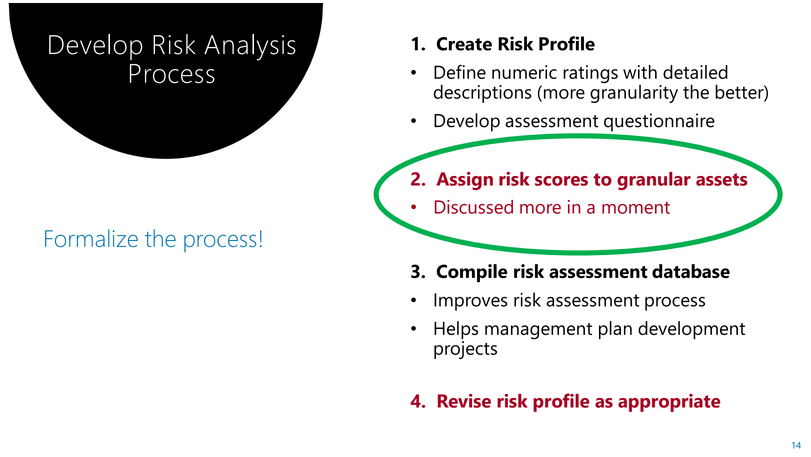
12



Benefits of Risk Analysis

1	2	3	4
Running tests in risk order	Allocating test effort based on risk	Allows organizations to make smarter release decisions	If schedule requires, drop test in reverse risk order
Find scary stuff first	Pick the right tests	Release when risk of delay balances risk of dissatisfaction	Give up tests you worry about the least

13



Develop Risk Analysis Process

Formalize the process!

1. **Create Risk Profile**
   - Define numeric ratings with detailed descriptions (more granularity the better)
   - Develop assessment questionnaire
2. **Assign risk scores to granular assets**
   - Discussed more in a moment
3. **Compile risk assessment database**
   - Improves risk assessment process
   - Helps management plan development projects
4. **Revise risk profile as appropriate**

14

14

Assigning Risk Scores

1

Assemble your list of assets (requirements or tests)

2

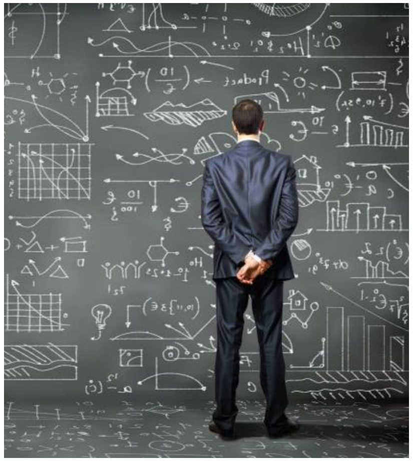
For each asset, determine the impact if the risk eventuates

3

For each asset, determine the likelihood the risk will eventuate

4

Calculate the Risk Score: a combination of the risk impact & risk likelihood & (perhaps) weight



Risk Analysis

IMPACT

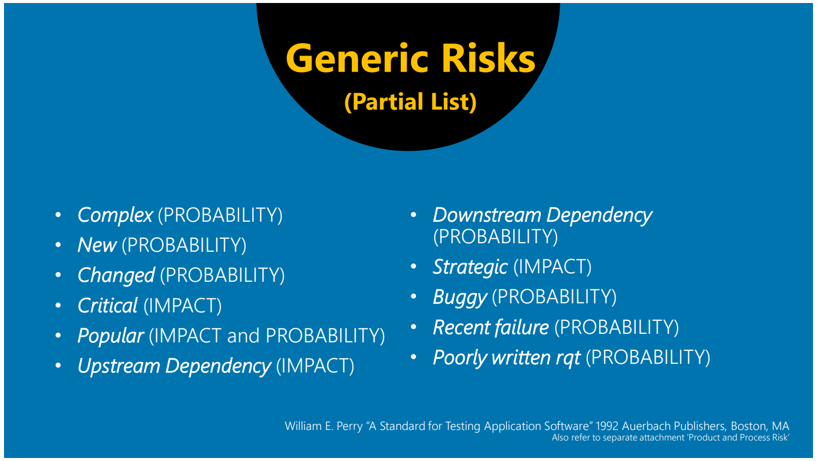
Loss of life?
Loss of revenue?
Inconvenience?
Exposure/frequency?

PROBABILITY

Is it *new* functionality or new technology?
Is it *existing* functionality? Has it been tested before?
Is it *mature* functionality?

WEIGHT(optional)

Additional factor(s) taken into account and factored into calculation to more adequately determine risk score



Generic Risks

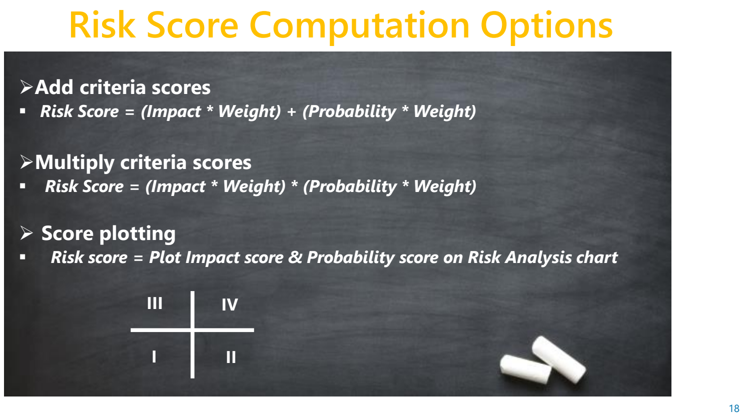
(Partial List)

- *Complex* (PROBABILITY)
- *New* (PROBABILITY)
- *Changed* (PROBABILITY)
- *Critical* (IMPACT)
- *Popular* (IMPACT and PROBABILITY)
- *Upstream Dependency* (IMPACT)
- *Downstream Dependency* (PROBABILITY)
- *Strategic* (IMPACT)
- *Buggy* (PROBABILITY)
- *Recent failure* (PROBABILITY)
- *Poorly written rqt* (PROBABILITY)

William E. Perry "A Standard for Testing Application Software" 1992 Auerbach Publishers, Boston, MA

Also refer to separate attachment 'Product and Process Risk'

17



Risk Score Computation Options

➢**Add criteria scores**

- ***Risk Score = (Impact * Weight) + (Probability * Weight)***

➢**Multiply criteria scores**

- ***Risk Score = (Impact * Weight) * (Probability * Weight)***

➢ **Score plotting**

- ***Risk score = Plot Impact score & Probability score on Risk Analysis chart***

18

18

Score Plotting Procedure

Three steps to scoring an application:

1

Determine the *Impact* of failure

2

Calculate the *Probability* score

3

Plot the scores on the Risk Analysis chart

Use results to focus test effort:

1st

Focus on components in Quadrant IV

2nd

Focus on components in Quadrant III

3rd

Focus on components in Quadrant II

4th

Focus on components in Quadrant I

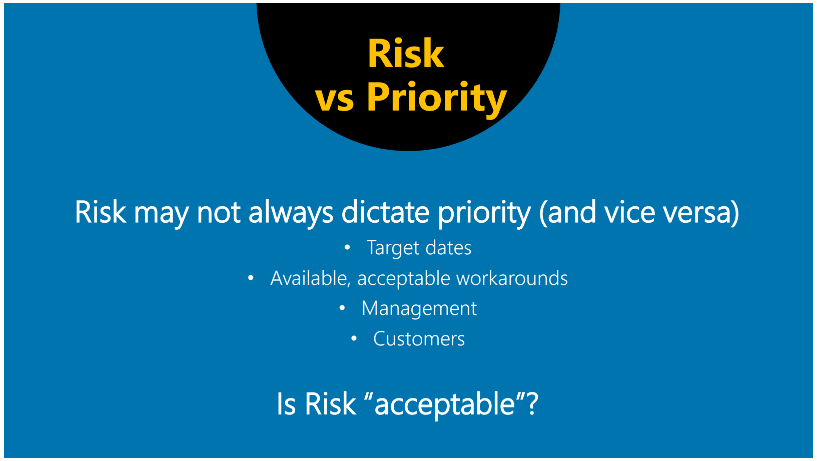
19

When would you consider a risk acceptable?

20

Besides risk, what other criteria might impact priority?

21



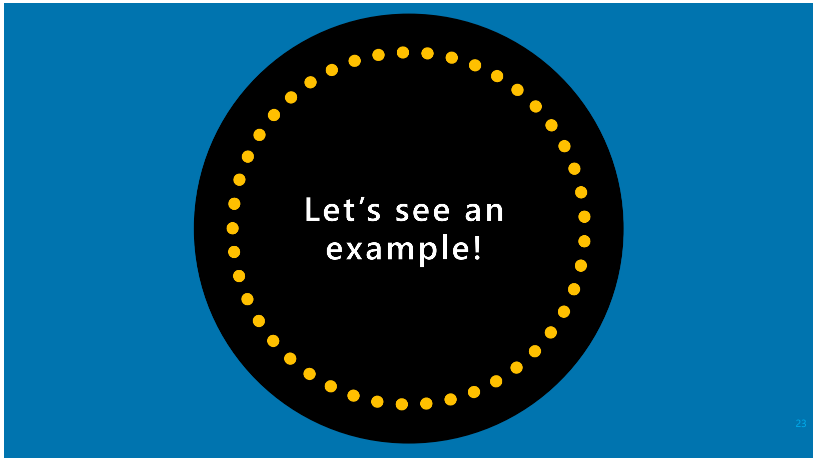
Risk vs Priority

Risk may not always dictate priority (and vice versa)

- Target dates
- Available, acceptable workarounds
- Management
- Customers

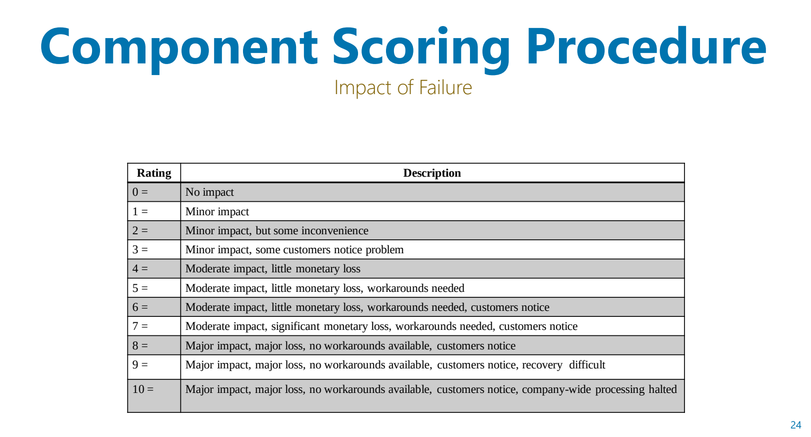
Is Risk “acceptable”?

22



Let’s see an example!

23

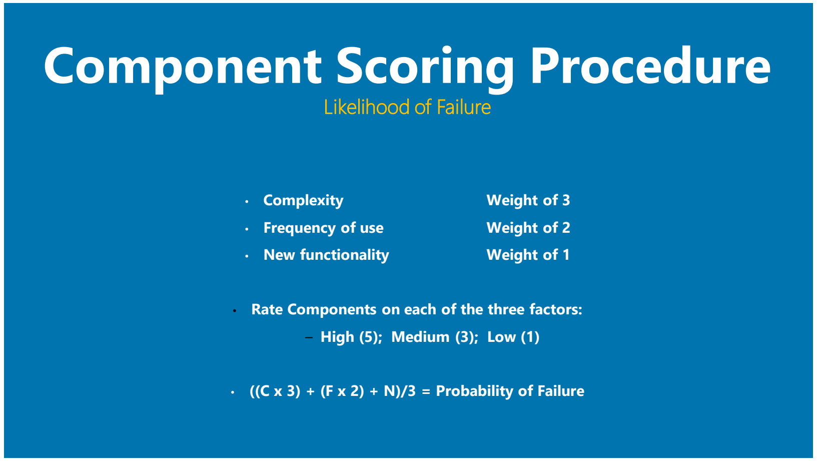


Component Scoring Procedure

Impact of Failure

Rating	Description
0 =	No impact
1 =	Minor impact
2 =	Minor impact, but some inconvenience
3 =	Minor impact, some customers notice problem
4 =	Moderate impact, little monetary loss
5 =	Moderate impact, little monetary loss, workarounds needed
6 =	Moderate impact, little monetary loss, workarounds needed, customers notice
7 =	Moderate impact, significant monetary loss, workarounds needed, customers notice
8 =	Major impact, major loss, no workarounds available, customers notice
9 =	Major impact, major loss, no workarounds available, customers notice, recovery difficult
10 =	Major impact, major loss, no workarounds available, customers notice, company-wide processing halted

24



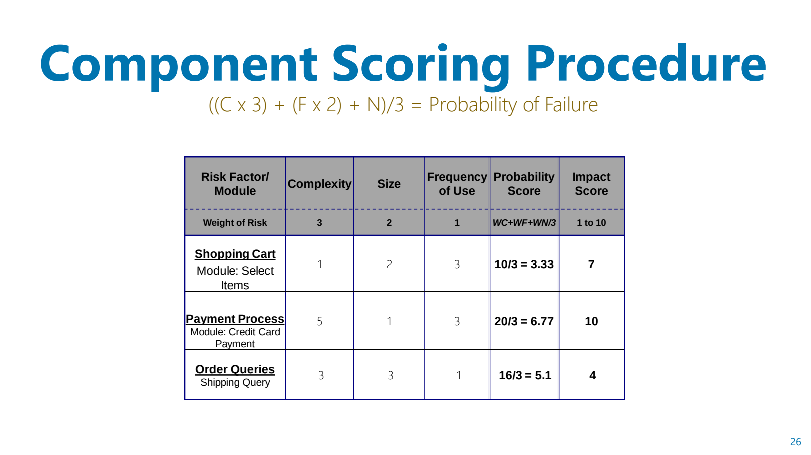
Component Scoring Procedure

Likelihood of Failure

Complexity	Weight of 3
Frequency of use	Weight of 2
New functionality	Weight of 1

- **Rate Components on each of the three factors:**
  - **High (5); Medium (3); Low (1)**
- **((C x 3) + (F x 2) + N)/3 = Probability of Failure**

25



Component Scoring Procedure

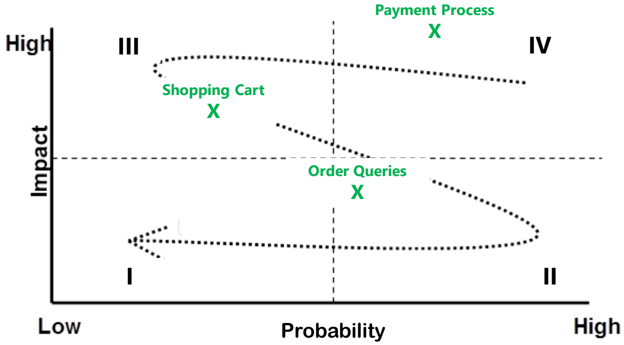
((C x 3) + (F x 2) + N)/3 = Probability of Failure

Risk Factor/ Module	Complexity	Size	Frequency of Use	Probability Score	Impact Score
Weight of Risk	**3**	**2**	**1**	***WC+WF+WN/3***	**1 to 10**
Shopping Cart Module: Select Items	1	2	3	**10/3 = 3.33**	**7**
Payment Process Module: Credit Card Payment	5	1	3	**20/3 = 6.77**	**10**
Order Queries Shipping Query	3	3	1	**16/3 = 5.1**	**4**

26

26

Component Scoring Procedure

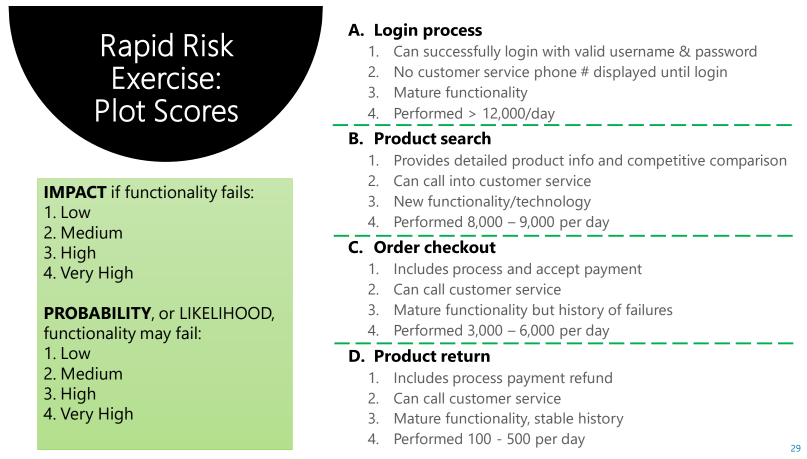


27

Rapid Risk Scoring

1. Get team or designated group together
2. Each individual gets card with scores of 1 (Low), 2 (Medium), 3 (High), and 4 (Very High). Can go more granular with wider range
3. Describe entity with same type of description for each. (Is it new? What's frequency? Exposure? Has it been a problem in the past?)
4. Give each person 5 seconds to hold up score card for Impact…for Probability
5. Average scores
6. Compute risk score – add, multiply, or plot

28



Rapid Risk Exercise: Plot Scores

IMPACT if functionality fails:
1. Low
2. Medium
3. High
4. Very High

PROBABILITY, or LIKELIHOOD, functionality may fail:
1. Low
2. Medium
3. High
4. Very High

A. Login process
1. Can successfully login with valid username & password
2. No customer service phone # displayed until login
3. Mature functionality
4. Performed > 12,000/day

B. Product search
1. Provides detailed product info and competitive comparison
2. Can call into customer service
3. New functionality/technology
4. Performed 8,000 – 9,000 per day

C. Order checkout
1. Includes process and accept payment
2. Can call customer service
3. Mature functionality but history of failures
4. Performed 3,000 – 6,000 per day

D. Product return
1. Includes process payment refund
2. Can call customer service
3. Mature functionality, stable history
4. Performed 100 - 500 per day

29

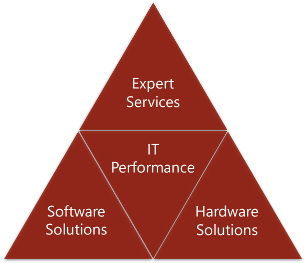
29

Questions?

30

About Checkpoint Technologies

- ✓ Incorporated in January, 2003
- ✓ QA and QC expertise focused on functional, performance and application security testing
- ✓ Micro Focus Software Platinum Partner, Authorized Software Support Partner & Certified Training Partner
- ✓ Atlassian Solution Partners
- ✓ Also partners with Mobile Labs, Kobiton, and QASymphony
- ✓ QAI Worldwide Training Partner



31

Contact me anytime!

Bob Crews

bcrews@checkpointech.com

Office: 813-818-8324 ext. 101

Direct: 813-493-3678

32

THANK YOU!

33