

2. Prioritisation and risk analysis

Prioritisation
Likelihood and impact estimation
Risk number
Planning risk and contingencies

The fundamental problem of testing software

- We cannot test for everything
- No system can be completely tested
- The need to have a clever testing methodology
- One solution: Test prioritisation based on risk analysis

2

"What you test is more important than how much you test."

Craig and Jaskiel

3

Risk driven testing

- Tests need to be prioritised so that the most important bugs are found first
- Risk driven testing is the recommended approach
- A dictionary definition of risk (Oxford Concise)

Hazard, chance of bad consequences or loss

4

Risk driven testing

Hazard, chance of bad consequences or loss

The probability, or likelihood, of an event occurring

The negative consequences or impact of the event

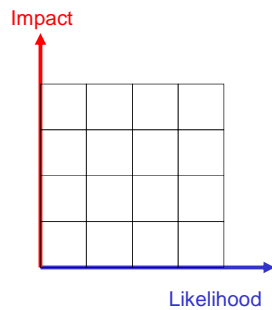
5

The components of risk

- **Likelihood**
An estimate of how probable is that the piece of a system would fail
- **Impact**
What would happen if this piece malfunctioned

6

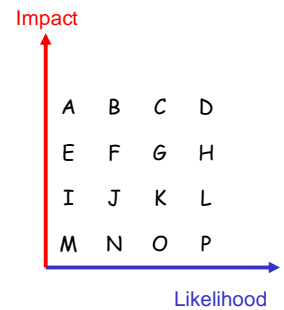
The components of risk



7

Prioritisation

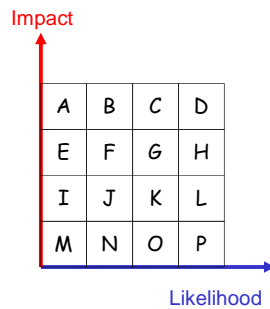
- Identify components
- Estimate the likelihood
- Estimate the impact
- Assign numerical values
- Prioritise the tests



8

Prioritisation

- Identify components
- Estimate the likelihood
- Estimate the impact
- Assign numerical values
- Prioritise the tests



9

Likelihood estimation

"On the scale 1 to 10, how shaky do you think this piece is?"

- Indicators
 - Complexity
 - Number of interfaces
 - Project history (new or modifications)
 - Development team history
 - ...
- Who?

10

Impact estimation

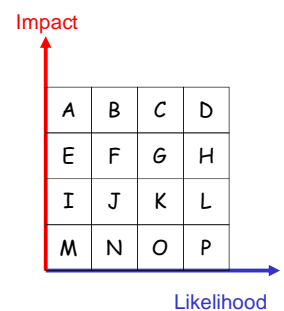
"How bad would it be, on a scale of 1 to 10, if this piece did not work correctly?"

- Indicators?
 - Application dependent
- Who?

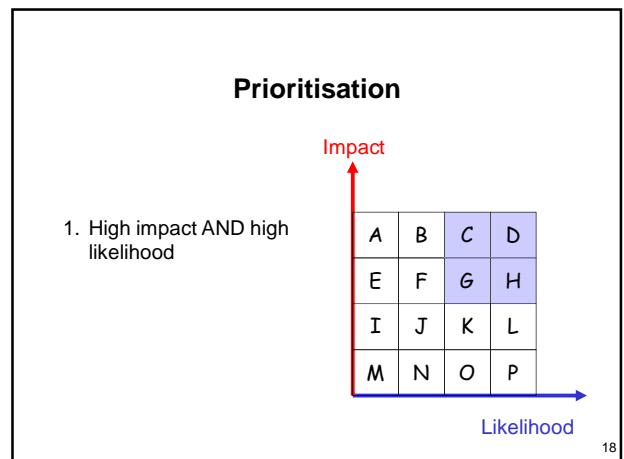
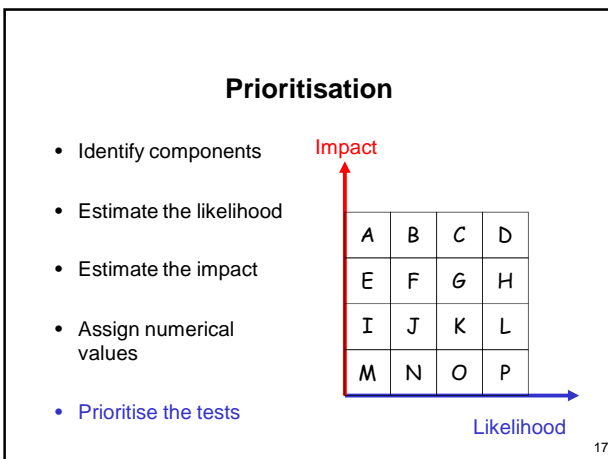
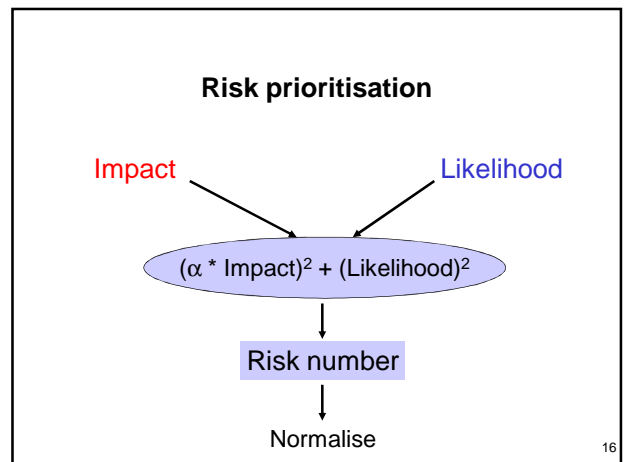
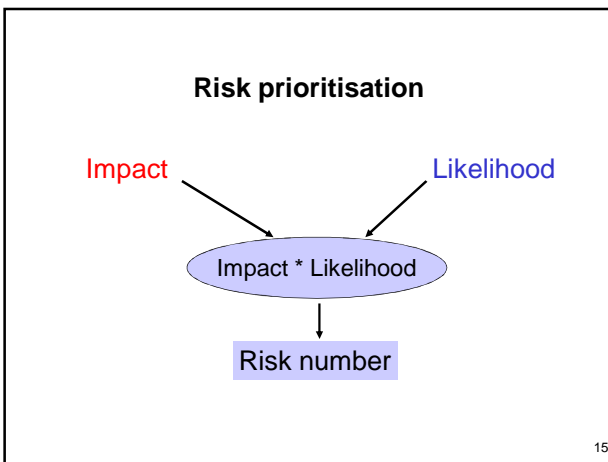
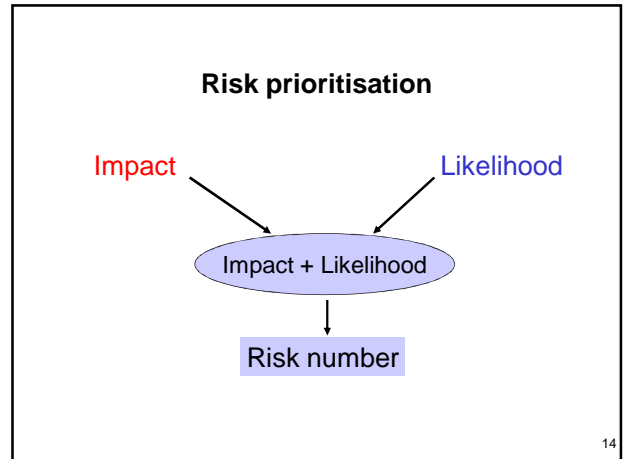
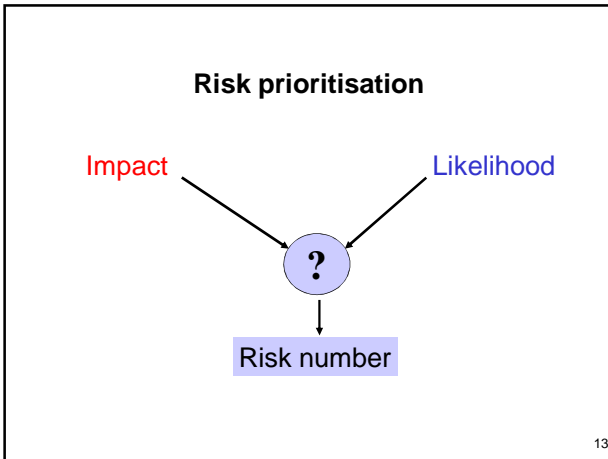
11

Prioritisation

- Identify components
- Estimate the likelihood
- Estimate the impact
- Assign numerical values
- Prioritise the tests

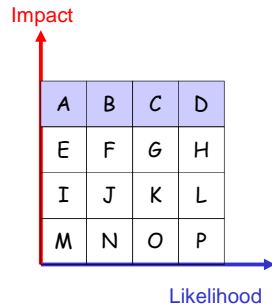


12



Prioritisation

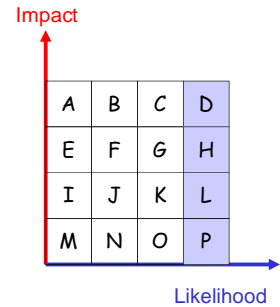
1. High impact AND high likelihood
2. High impact



19

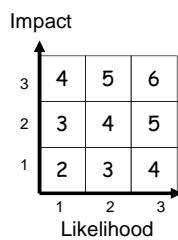
Prioritisation

1. High impact AND high likelihood
2. High impact
3. High likelihood

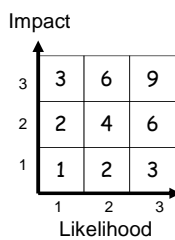


20

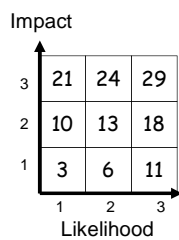
Examples



$$RN = I + L$$



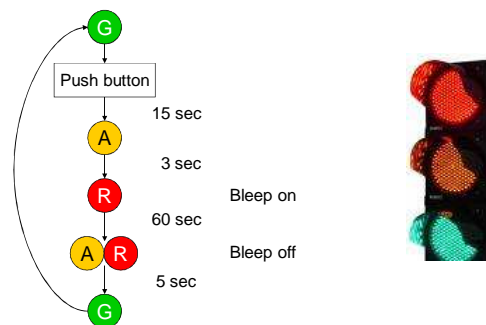
$$RN = I * L$$



$$RN = (\alpha^* I)^2 + (L)^2$$
$$\alpha = 1.5$$

21

EXERCISE: traffic light system



22

EXERCISE: traffic light system

- Features
 - Push button
 - Get time
 - Green on
 - Green off
 - Amber on
 - Amber off
 - Red on
 - Red off
 - Bleep on
 - Bleep off

23

EXERCISE: traffic light system



	Feature	Likelihood of failure (1-10)	Impact of failure (1-10)	Risk number
F1	Push button			
F2	Get time			
F3	Green on			
F4	Green off			
F5	Amber on			
F6	Amber off			
F7	Red on			
F8	Red off			
F9	Bleep on			
F10	Bleep off			

24 |

Risk analysis: WHO?

- A broad team
 - Users
 - Developers
 - Testers
 - Support personnel
 - Marketing team
 - Application domain experts

26

Risk analysis: HOW?

- Brainstorming sessions
- Based on all available sources of information
 - Requirement specification
 - Functional specification
 - Defect reports
 - User experience
 - Developer / Tester experience
 - ...
- First collect ideas – no debating

27

Risk analysis: HOW?

- Examples of things that can go wrong
 - "Real world" effects
 - Computer problems
 - Incorrect input
 - Failure to meet user expectations
 - ...

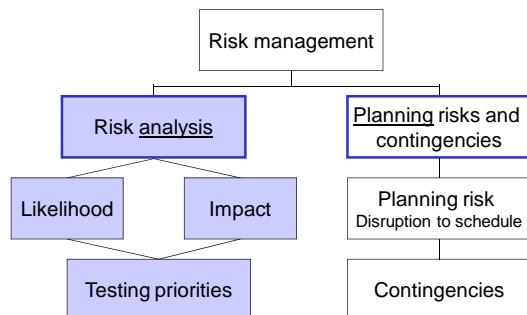
28

Risk analysis based on Requirement Specification

- **Assumptions**
- **Domain knowledge** – need continuous verification v. customer
- **Risk factors** – how complex is this requirement? How mature is technology? Does the team has capability?
- **Priority / release date** – when is this requirement needed?
- **Work planning** – what work is needed to test this requirement?
- **Benefits** – what business benefit does this requirement contribute to?
- **Satisfaction / dissatisfaction factor** – how happy / unhappy will the customer be if this requirement is met / not met

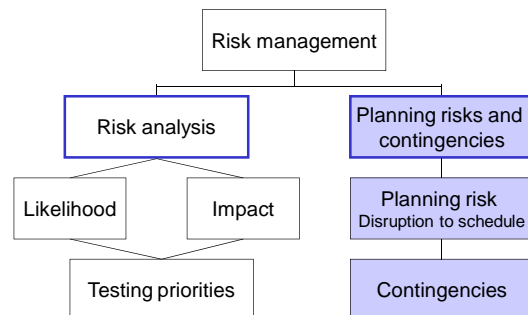
From "A Primer on Requirements Engineering", Beaver Consulting²⁹

Risk management



31

Risk management



32

Planning risks and contingencies

"Murphy's Law is alive and well"

- Planning risk examples
 - Late requirements
 - Late delivery of software
 - Computer environment problems
 - ...
- Use "What if ..." analysis to develop contingencies

33

Planning risks and contingencies

- For the high impact components a full contingency plan is necessary
- Examples of contingencies
 - Reduce the scope of the project
 - Delay implementation
 - Add resources
 - Reduce quality process
 - ...

34

RISK	HOW?	TESTS NEEDED	CONTINGENCY PLAN
No access	- System down - Database unavailable	ST12.34 – 12.75 ST45.12, ST31.1	Mirror system
Wrong value	- Old value used - conversion factor not applied	ST12.34 ST45.12	Allow manual entry of value

35

Homework



In the "IEEE Standard 829-1998 for Software Test Documentation"
http://www.cs.bham.ac.uk/~exc/Teaching/STesting/Web_resources.html

- Identify item "Risk and contingencies" in the outline test plan (4.2)
- Study an example of the Risk and contingencies plan (p. 26).

36

Further reading



- Craig and Jaskiel "Systematic Software Testing", Chapter 2: "Risk analysis"
- "A Primer on Requirements Engineering", Beaver Consulting
http://www.cs.bham.ac.uk/~exc/Teaching/STesting/Requirement_Primer.pdf

37

Next lecture

Practical example in test case design

38