

# COMPX201/Yo5335

## Data Structures and Algorithms



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*

Credits: Jemma König (UoW)

# An Introduction to Testing

COMPX201/Yo5335

# Overview

- The importance of testing
- Types of testing
- Legal and ethical considerations

# The Importance of Testing

# The importance of testing

Why do we test software?

- Improve code quality
- Catch and fix bugs
- Improve user experience
- Find and fix faults in the system
- Mitigate risk of system failure

# The importance of testing

Why else do we test software?

- It is a predictable way to develop – you know when you are 'finished', without having to worry about a long bug trail
- It improves the lives of users
- It gives you a chance to learn all of the lessons that the code has to teach you. If you only slap together the first thing you think of, then you never have time to think of a second, better thing.

# The importance of testing

Why else do we test software?

- For large scale projects, 30-60% of the development effort may be taken up by testing
- A study conducted by NIST in 2002 reported that software bugs cost the U.S. economy \$59.5 billion annually. More than a third of this cost could be avoided if better software testing was performed

# The importance of testing

Why else do we test software?

- Radiation therapy machines that have maimed or killed people
- Ongoing computer system problems in a health insurance company resulted in denial of coverage for needed medications and mistaken overcharging or cancellation of benefits
- In 2008, more than 600 U.S. airline flights were significantly delayed due to a software glitch in the U.S. FFA air traffic control system

[http://www.softwareqatest.com/qatfaq1.html#FAQ1\\_3](http://www.softwareqatest.com/qatfaq1.html#FAQ1_3)



# Types of Testing

# Types of testing

- Hazard and risk analysis
  - Preliminary Hazard Identification (PHI)
  - Failure Modes and Effects Analysis (FMEA)
  - Hazard and Operability studies (HAZOP)
  - Event Tree Analysis (ETA)
  - Fault Tree Analysis (FTA)
- Software testing
  - Black box & white box testing
  - Acceptance testing
  - Regression testing
  - Usability testing
  - Unit testing

# Types of testing

- Code-based testing
  - Code path coverage
  - Statement coverage
  - Branch coverage
  - Condition coverage
  - Mutation testing
  - Decision tables
- And a whole lot more ...
  - Load testing
  - Stress testing
  - And more ...

# Legal and Ethical Considerations

# Legal and ethical considerations

As the person who designs/builds/certifies a system, you may be prosecuted under:

- Criminal liabilities (from safety at work and consumer protection laws) e.g., negligence
- Civil liabilities (from contract law and sale of goods law)

# Legal and ethical considerations

To protect yourself from prison/fines, you might

- Ensure that your development practice is state of the art, e.g., you follow the appropriate guidelines and standards
- Carefully label and market your product to ensure safe usage and to transfer some responsibility to the user
- Insure yourself/your company against civil claims

# Hazard and risk analysis example

“A relatively new popular jet aircraft that had been in service for less than two years was grounded worldwide in March 2019 after two fatal air crashes, and production was suspended by the aircraft manufacturer in December 2019. Reportedly the crashes were due to a software flaw that caused serious problems when there was unexpected system input from sensor data. As of January 2020 the manufacturer was working on a fix and recertification, however it was also reported that a new software issue was found in the aircraft, likely further delaying recertification. It was also reported that the manufacturer had already lost billions of dollars in revenue, its stock price had plunged, it faced multiple lawsuits, and its CEO was replaced.”

[http://www.softwareqatest.com/qatfaq1.html#FAQ1\\_3](http://www.softwareqatest.com/qatfaq1.html#FAQ1_3)

# COMPX201/Yo5335

## Data Structures and Algorithms



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*

Credits: Jemma König (UoW)



# Hazard and Risk Analysis

COMPX201/Yo5335

# Overview

- Hazard and risk analysis
- Hazard analysis techniques

# Hazard and Risk Analysis

# Hazard and risk analysis

A hazard is a potential source of harm. In other words, hazards are areas where things could potentially go wrong, or where things could potentially lead to faults

Risk refers to the chance of a hazard causing harm

# Hazard and risk analysis example

"It was widely reported in August 2019 that a major North American bank had suffered a data breach exposing the data from more than 100 million credit card applicants, due to a misconfigured cloud service. Three months later news articles about the bank indicated that technical problems blocked customers' access to accounts and direct deposits for part of a day. In March of that year there were reports that the bank had an outage of its mobile and online banking services. Previously, in 2018 there were multiple reports of issues with the same bank - in February 2018 it was reported that 50GB of bank data was found to be publicly accessible due to an issue at one of the bank's vendors, and a month prior to that it was reported that the bank had an 'internal tech issue' causing accounts to have multiple charges for the same debit card transactions, resulting in unexpected negative balances and overdrafts. Some articles in 2019 stated concerns regarding 'reduced attention to basic software testing' and concerns regarding 'core software testing and maintenance'. In August of 2020 the bank was fined \$80 million by a government regulatory agency for '...failure to establish effective risk assessment processes...' and was required to '...develop appropriate risk mitigation testing from the beginning and throughout new project life cycle...'."

[http://www.softwareqatest.com/qatfaq1.html#FAQ1\\_3](http://www.softwareqatest.com/qatfaq1.html#FAQ1_3)

# Hazard and risk analysis

Hazards are areas where things could potentially go wrong, or where things could potentially lead to faults. Risk refers to the chance of a hazard causing harm

The goal of hazard and risk analysis is to identify the hazards of a system

- This allows us to determine the risk of those hazards
- And decide whether we need to redesign the system (to remove faults or reduce their effects)

# Hazard and risk analysis

- Faults are inevitable, because of:
  - Imperfect design
  - Random failure due to wear, aging, or other effects
- "*Engineering*" means finding acceptable compromises between goals like safety, correctness, cost, time-to-market, etc.

# Hazard analysis techniques

- Hazard analysis techniques include:
  - Preliminary Hazard Identification (PHI)
  - Failure Modes and Effects Analysis (FMEA)
  - Hazard and Operability studies (HAZOP)
  - Event Tree Analysis (ETA)
  - Fault Tree Analysis (FTA)

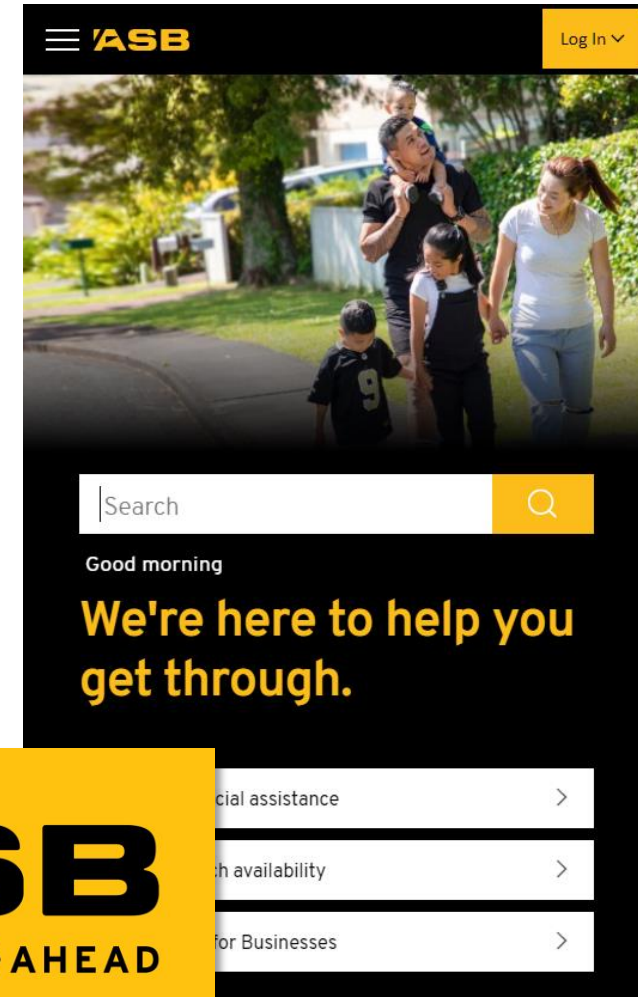


# Preliminary Hazard Identification (PHI)

# Preliminary Hazard Identification (PHI)

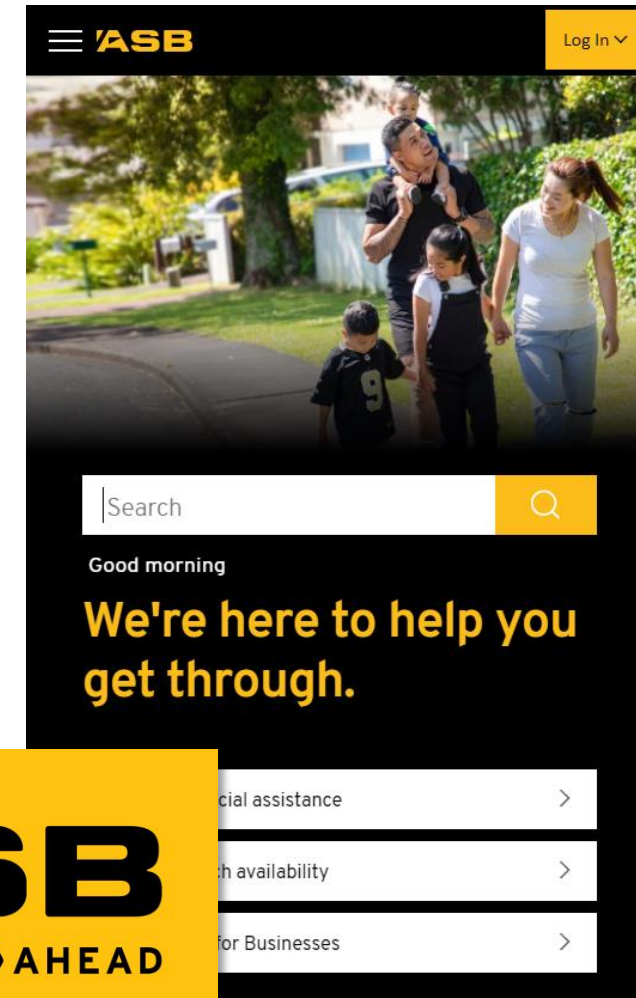
- Initial high-level screening
- Used to identify, describe, and rank major hazards
- Conducted during the conceptual stage
  
- PHI provides information on hazards early in the lifecycle
- Allows for further investigation into hazard mitigation
- BUT
  - This is not a systematic method
  - The quality of a PHI depends on the experience of the person/team conducting the analysis

# Preliminary Hazard Identification (PHI) example



# Preliminary Hazard Identification (PHI) example

Hazard	Potential fault	Rank
Data storage and security	A fault in data storage could result in a loss of user data, or a potential security breach	High
Network security	Data could be leaked during transfer over network	High
Network connectivity	An interruption in network connectivity could prevent users from accessing our services	Medium



**ASB**  
ONE STEP > AHEAD

# Failure Modes and Effects Analysis (FMEA)

# Failure Modes and Effects Analysis (FMEA)

- A table-based notation for considering the effects of failure of each component
- Very large and detailed for complex systems
- A weakness is that only single failures are considered
- An extension (FMECA) considers critically, which allows effort to be directed at the areas of greatest need

# Failure Modes and Effects Analysis (FMEA) example

Process Step	Potential Failure Mode	Potential Failure Effect	SEV <sup>1</sup>	Potential Causes	OCC <sup>2</sup>	Current Process Controls	DET <sup>3</sup>	RPN <sup>4</sup>	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	<ul style="list-style-type: none"> <li>Unauthorized cash withdrawal</li> <li>Very dissatisfied customer</li> </ul>	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	<ul style="list-style-type: none"> <li>Transaction failure</li> <li>Network issue</li> </ul>	3	Install load balancer to distribute work-load across network links	4	96	
	Extra cash dispensed	Bank loses money	8	<ul style="list-style-type: none"> <li>Bills stuck to each other</li> <li>Bills stacked incorrectly</li> </ul>	2	Verification while loading cash in ATM	3	48	

- 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.
- 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.
- 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.
- 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 and Operability studies (HAZOP)



# Hazard and Operability studies (HAZOP)

- Structured and systematic
- Identify potential hazards and operability problems that are likely to lead to non-conforming products
- Assumes risk events are caused by deviations from design
- Uses a set of 'guide words' to identify deviations
- Focuses on flows of data and material through a system

# Hazard and Operability studies (HAZOP)

- HAZOP is often described as:
  - A brainstorming technique
  - A qualitative risk assessment tool
  - An inductive risk assessment tool, meaning that it is 'bottom up'
- Success relies on subject-matter-experts (SME) predicting deviations based on past experiences and general subject matter experience

[https://pqri.org/wp-content/uploads/2015/08/pdf/HAZOP\\_Training\\_Guide.pdf](https://pqri.org/wp-content/uploads/2015/08/pdf/HAZOP_Training_Guide.pdf)

# Hazard and Operability studies (HAZOP)

Risk assessment teams are responsible for collating a set of guide words. Some common HAZOP guide words include:

- |                              |   |
|------------------------------|---|
| ▪ <b>No or not</b>           | ▪ <b>Other than</b>                         |
| ▪ <b>More</b>                | ▪ <b>Early</b>                              |
| ▪ <b>Less</b>                | ▪ <b>Late</b>                               |
| ▪ <b>As well as</b>          | ▪ <b>Before</b>                             |
| ▪ <b>Part of</b>             | ▪ <b>After</b>                              |
| ▪ <b>Reverse (of intent)</b> | ▪ <b>Others can be crafted as needed...</b> |

# Hazard and Operability studies (HAZOP)

Risk assessment teams are responsible for collating a set of guide words. Some common HAZOP guide words include:

- |  |  |
|--|--|
| ▪ <b>No or not</b> - no detergent added  | ▪ <b>Part of</b> - critical detergent component omitted (ex: surfactant)   |
| ▪ <b>More</b> - too much detergent volume added (difficult to rinse)               | ▪ <b>Reverse</b> - detergent is contaminated with a harmful hazard   |
| ▪ <b>More</b> – supplied detergent solution concentration is too high              | ▪ <b>Other than</b> - wrong detergent used   |
| ▪ <b>Less</b> - too little detergent volume added (soil isn't effectively removed) | ▪ <b>Early</b> - detergent added too early (ex: if you need to pre-rinse bulk soil to drain before washing with detergent) |
| ▪ <b>Less</b> – supplied detergent solution concentration is too low               | ▪ <b>Late</b> - detergent added too late in the cleaning cycle   |

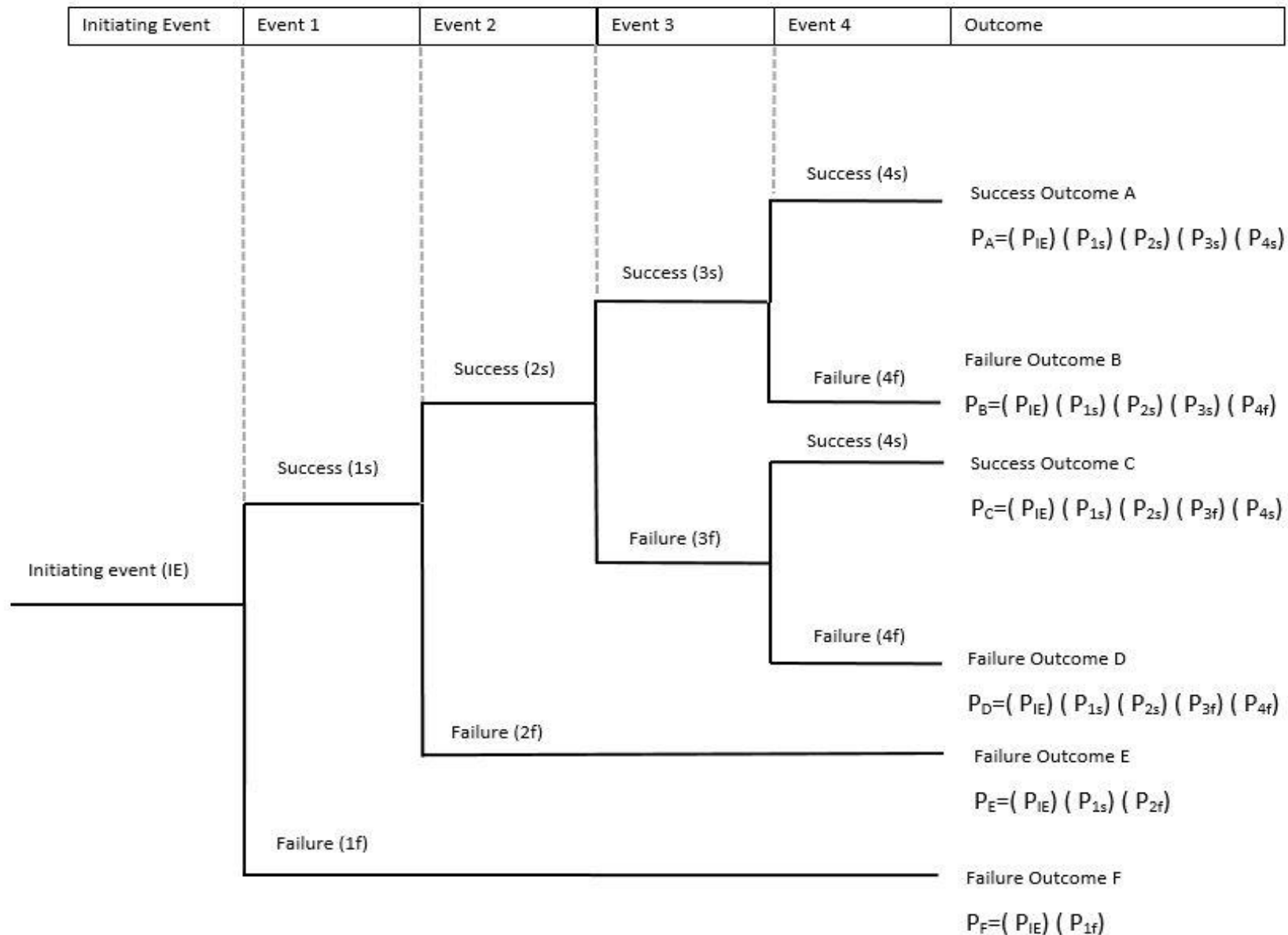
# Event Tree Analysis (ETA)

# Event Tree Analysis (FTA)

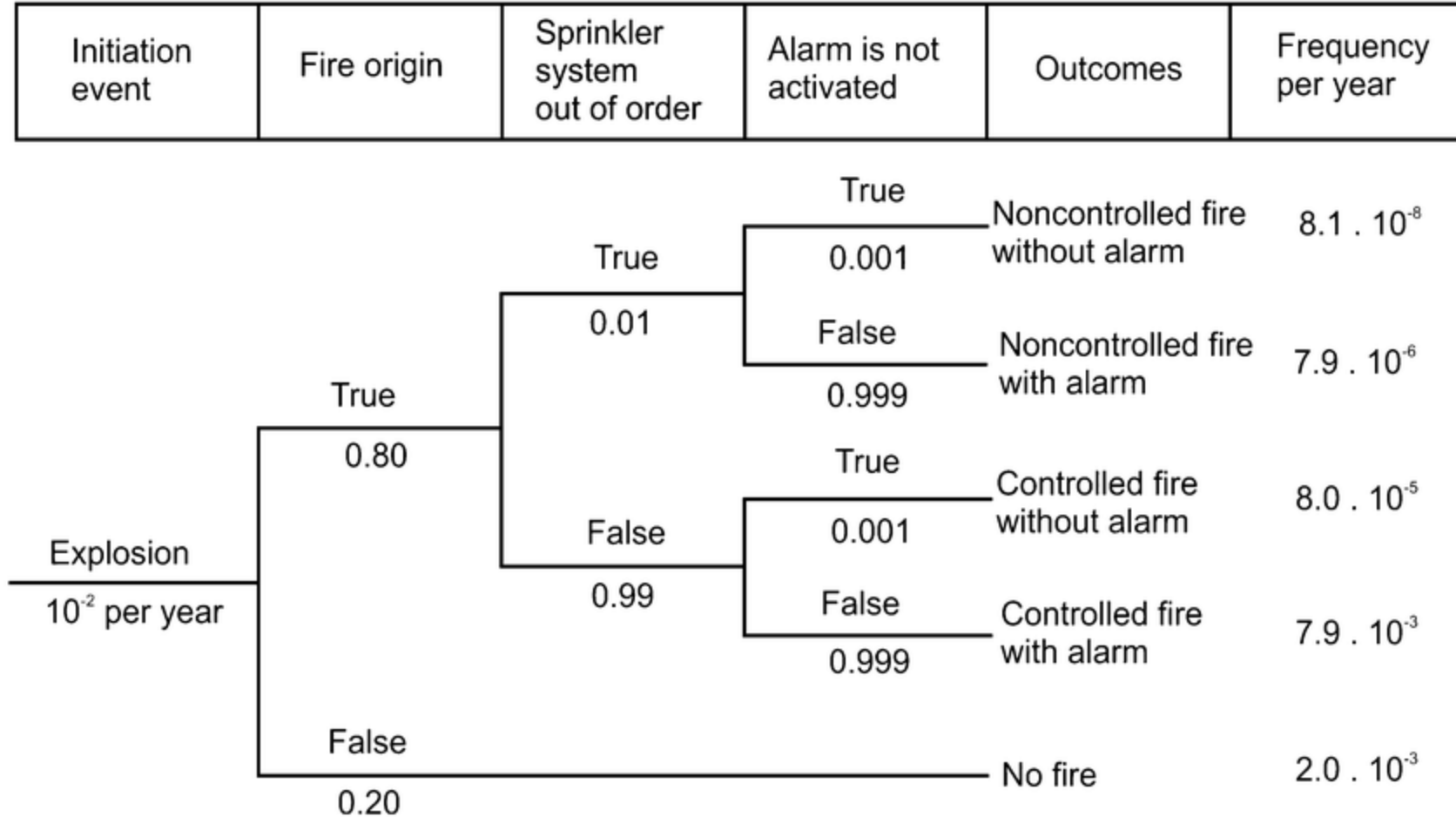
- ETA is a tree notation used to follow the effects of an event forward through a system
- ETA can be used to determine the probability of positive and negative outcomes occurring
- Starts with an 'initiating event'
- Consequences of this event follow in a binary success/failure structure
- Each event creates a path where a series of successes/failures occur
- For example ...

[https://en.wikipedia.org/wiki/Event\\_tree\\_analysis](https://en.wikipedia.org/wiki/Event_tree_analysis)

# Event Tree Analysis (FTA)



# Event Tree Analysis (FTA) example



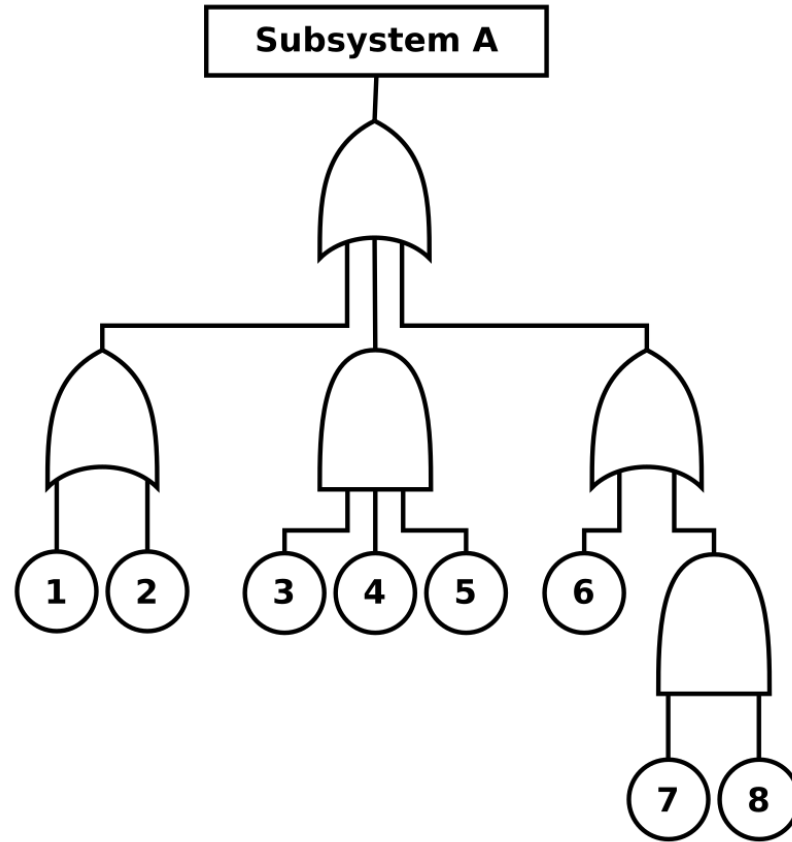


# Fault Tree Analysis (FTA)

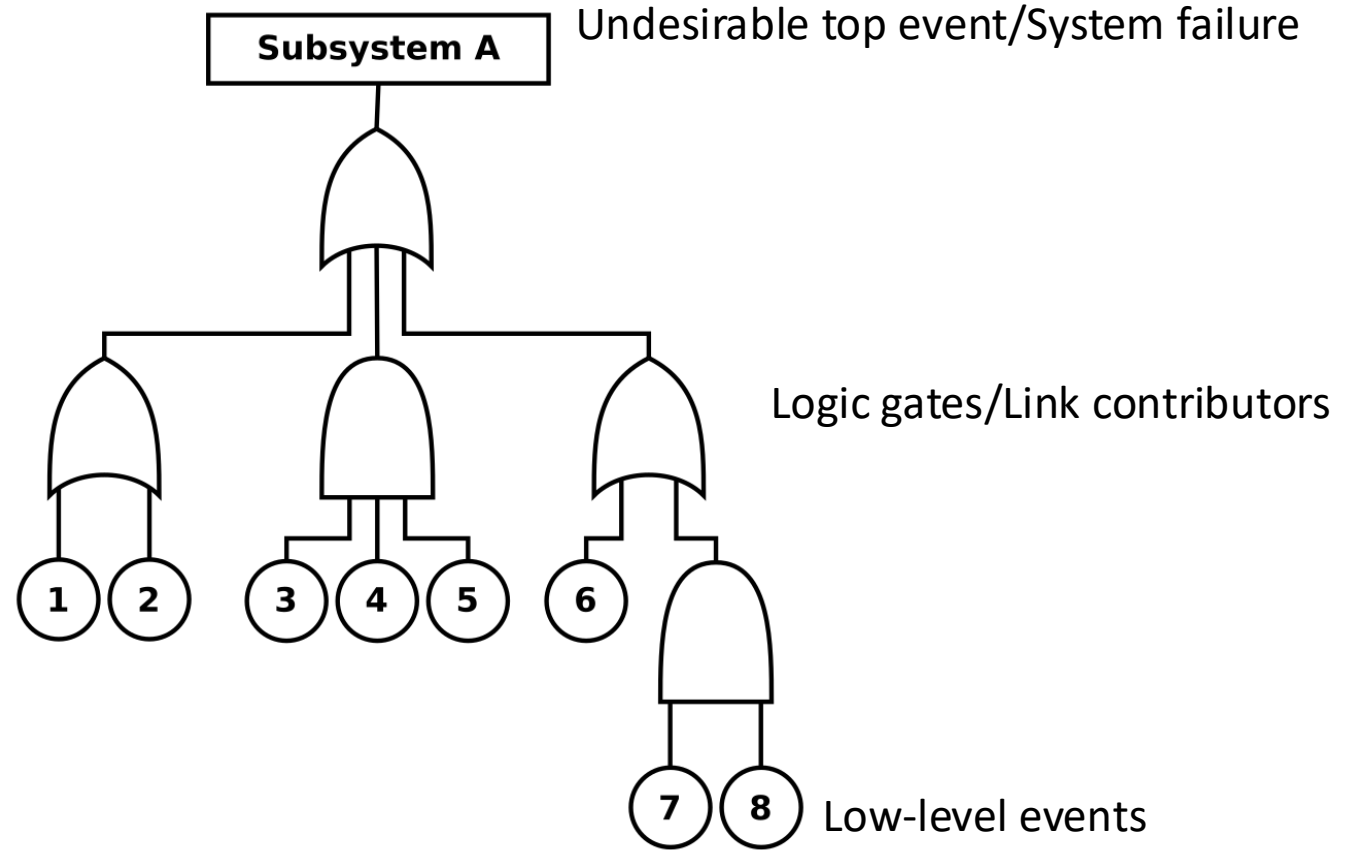
# Fault Tree Analysis (FTA)

- FTA is a tree notation for following the causes of hazards back through a system
- Used to explore the causes of system level failures
- Uses Boolean logic to combine low-level event to identify component level failures
- Links component level failures back to system level failures
- For example ...

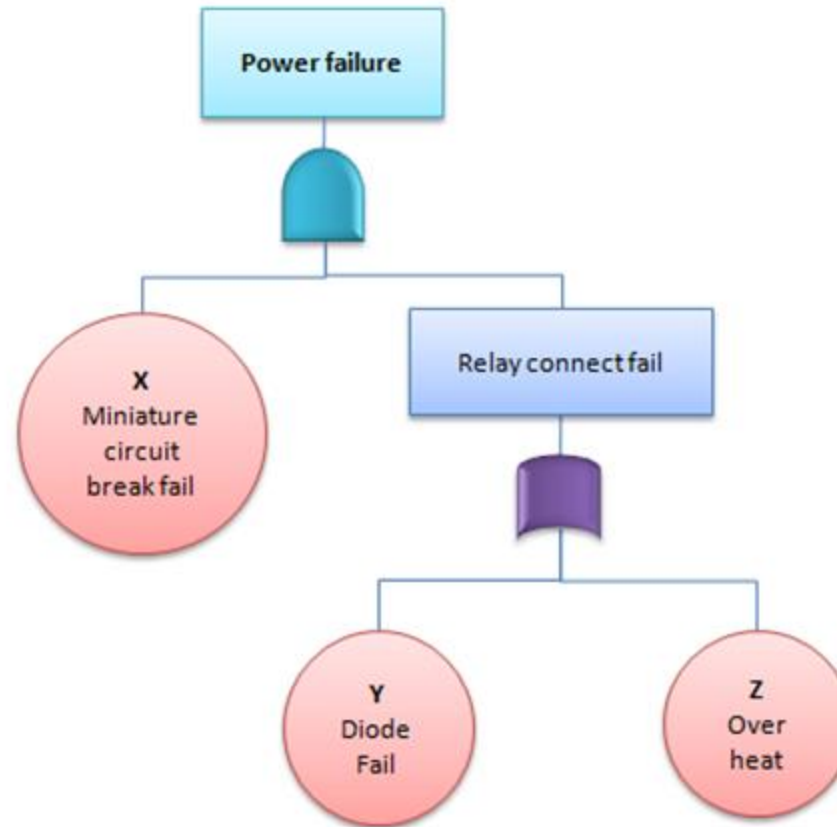
# Fault Tree Analysis (FTA)



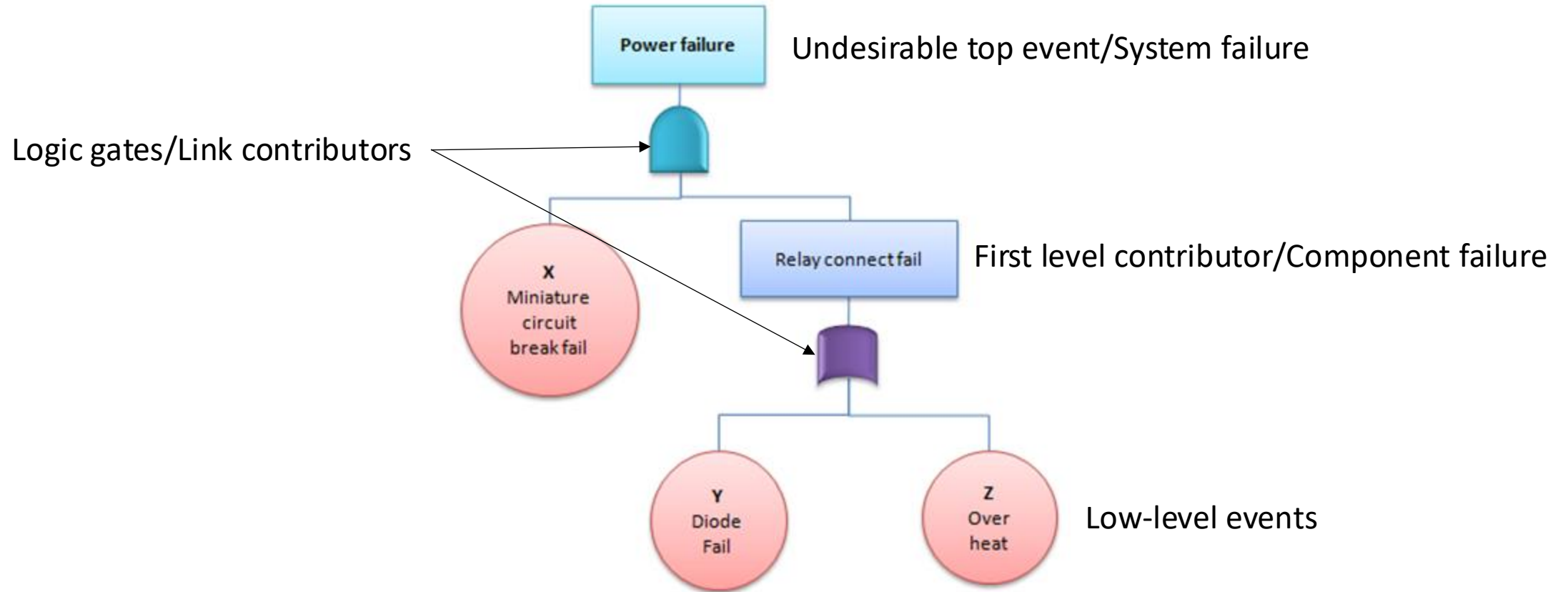
# Fault Tree Analysis (FTA)



# Fault Tree Analysis (FTA) example










# Fault Tree Analysis (FTA) example







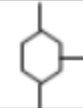
# Fault Tree Analysis (FTA)

- Event symbols:

S.No	Event Symbol	Description
1		Primary or basic failure event. It is a random event and sufficient data is available
2		State of system, subsystem or component event
3		Secondary failure or under developed event, can be explored further
4		Conditional event and is associated with the occurrence of some other event
5		House event representing either occurrence or non-occurrence of an event
6	 In  Out	Transfer in and transfer out symbols used to replicate a branch or sub-tree of the FTA

# Fault Tree Analysis (FTA)

- Gate symbols:

S.No	Gate Symbol	Description
1	 AND Gate	The output event occurs when all the input events occur
2	 OR Gate	The output event occurs when at least one of the input events occur
3	 Priority AND Gate	The output event occurs when all the input events occur in the order from left to right
4	 Exclusive OR gate	The output event occurs if either of the two input events occur but not both
5	 Inhibit gate	The output event occurs when the input event occurs and the attached condition is satisfied



# COMPX201/Yo5335

## Data Structures and Algorithms



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*

Credits: Jemma König (UoW)

# Measuring Risk

COMPX201/Yo5335

# Overview

- Risk
- Techniques for reducing risk
- Measuring risk

# Risk

Risk refers to the chance of a hazard causing harm, and the severity of that harm

# Techniques for reducing risk

- Fault avoidance:
  - Aim to prevent faults from entering the system during design. E.g., Hazard analysis, inspections, formal methods.
- Fault removal:
  - Attempt to find faults within a system before it enters service. E.g., Testing
- Fault detection:
  - Used during service to detect faults and minimize their effects. E.g., Maintenance
- Fault tolerance:
  - Allow the system to operate correctly in the presence of faults. E.g., Redundancy

# Measuring Risk

# Measuring risk

- Accident:  
An unintended event or sequence of events that cause death, injury, environmental or material damage
- Hazard:  
A situation in where there is actual or potential danger to people or to the environment
- Fault:  
A defect in the system

# Measuring risk

## Severity

Refers to how major the consequences of an accident are.

1. Catastrophic:  
Multiple deaths
2. Critical:  
A single death, and/or multiple severe injuries or severe occupational illnesses
3. Marginal:  
A single severe injury or occupational illness, and/or multiple minor injuries or minor occupational illnesses
4. Negligible:  
At most a single minor injury or minor occupational illness



# Measuring risk

## Frequency

How likely an accident is to happen. E.g., 2 accidents/year,  $10^{-9}$  safety incidents/operating-hour,  $10^{-7}$  incidents/use. Categories are industry-specific, but typically look something like the following

1. Frequent:  
Likely to be continually experienced
2. Probable:  
Likely to occur often
3. Occasional:  
Likely to occur sometimes
4. Improbable:  
Unlikely, but may exceptionally occur
5. Incredible:  
Extremely unlikely that the event will occur at all

# Measuring risk

$$\textit{risk} = \textit{severity} \times \textit{frequency}$$

# Measuring risk

Frequency	Severity			
	Catastrophic	Critical	Marginal	Negligible
Frequent	1	1	1	2
Probable	1	1	2	3
Occasional	1	2	3	3
Remote	2	3	3	4
Improbable	3	3	4	4
Incredible	4	4	4	4

# Measuring risk

Frequency	Severity			
	Catastrophic	Critical	Marginal	Negligible
Frequent	1	1	1	2
Probable	1	1	2	3
Occasional	1	2	3	3
Remote	2	3	3	4
Improbable	3	3	4	4
Incredible	4	4	4	4

Where:

1. = Intolerable risk

2. = Undesirable risk

Tolerable only if reduction is impracticable or if the costs are grossly disproportionate to the improvement gained

3. = Tolerable risk

Tolerable if the cost of risk reduction would exceed the improvement gained

4. = Negligible risk

# Finally, please note

- For software in particular ...
- ... everyone knows that software will inevitably contain errors or mistakes, or faults.
- Software does not fail randomly or degrade with age. All faults are systematic and due to poor design/implementation/etc.
- Due to the complexity of most software, exhaustive testing is impossible
- Instead, we implement testing strategies and use testing software ...
- ... more on this later

# COMPX201/Yo5335

## Data Structures and Algorithms



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*

Credits: Jemma König (UoW)