

INFO1111: Computing 1A Professionalism

Week 6: Systems Thinking Problem Solving



THE UNIVERSITY OF
SYDNEY

“If I had an hour to solve a problem I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions.”

— *Albert Einstein*

“The only way to know how a complex system will behave after you modify it, is to modify it and see how it behaves”.

— *George E. P. Box*

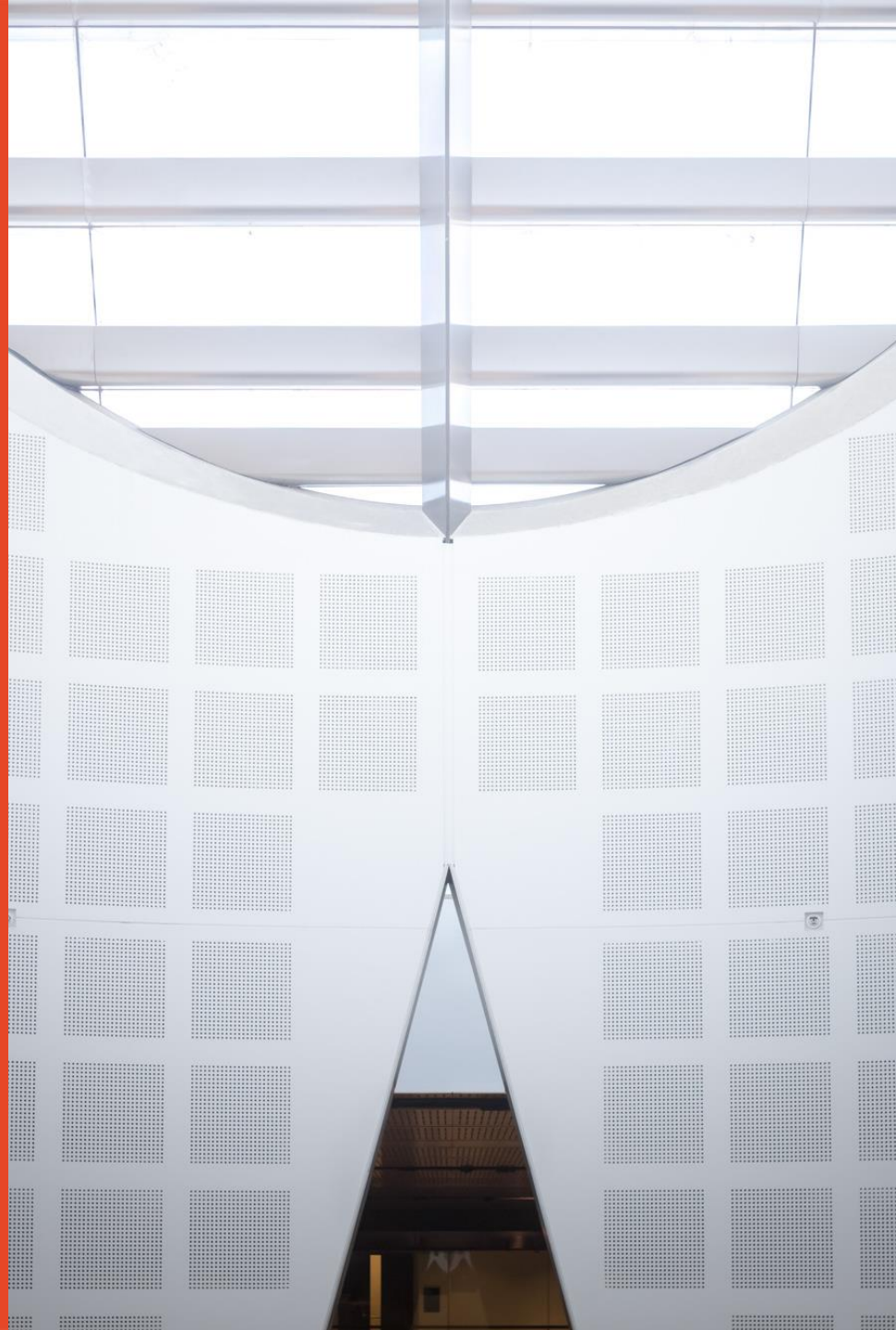
INFO1111: Computing 1A Professionalism

Week 6: Systems Thinking Problem Solving

Week 5 recap
Self-Learning
Week 6 overview



THE UNIVERSITY OF
SYDNEY

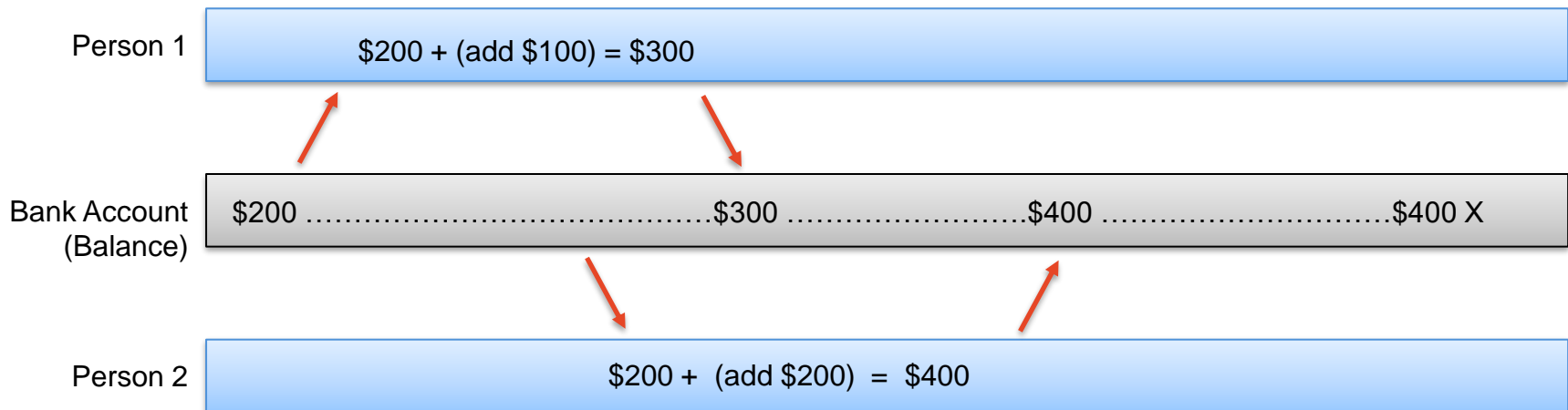


Week 5 Recap

- Finding Information
 - How good is your Google Fu?
 - How do you improve it?
- Referencing
- Version Control
 - File naming – only works in simple cases!
 - Versions aren't just “linear”
 - We have different branches...
- Git
 - This is a *key* skill. If you are struggling, then don't just avoid it..

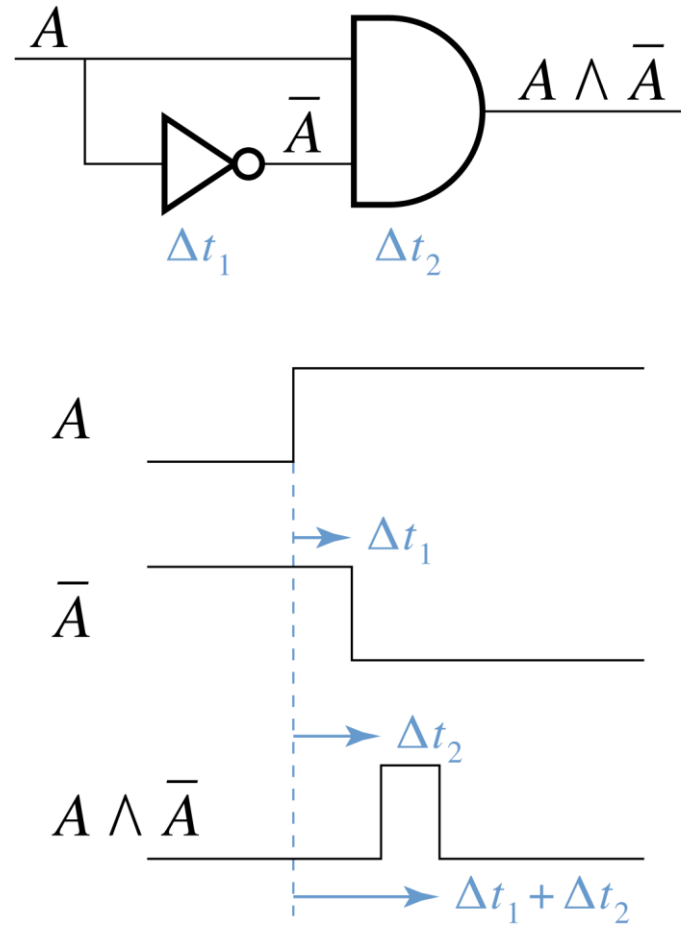
Week 5 self-learning exercise...

- Week 5 Concept: Race Condition
 - What is a race condition?
 - Under what circumstances might it be important?
- Where event timing can have an impact on the outcome...
 - Many applications / programs are “multi-threaded”
 - Interactions between these threads can cause ‘Heisenbugs’
- Consider the following (2 people depositing money into 1 account)



Week 5 self-learning exercise...

- It can happen in hardware too!



Week 6 self-learning exercise...

- What is Hill Climbing?
 - (nothing to do with hiking!)

Week 6 Overview

- Systems (and Systems thinking)
 - Most software is part of complex systems!
- Problem solving
 - How good are you at solving problems?
 - What techniques do you use?
 - *Think back to the domino problem way back in week 1...*

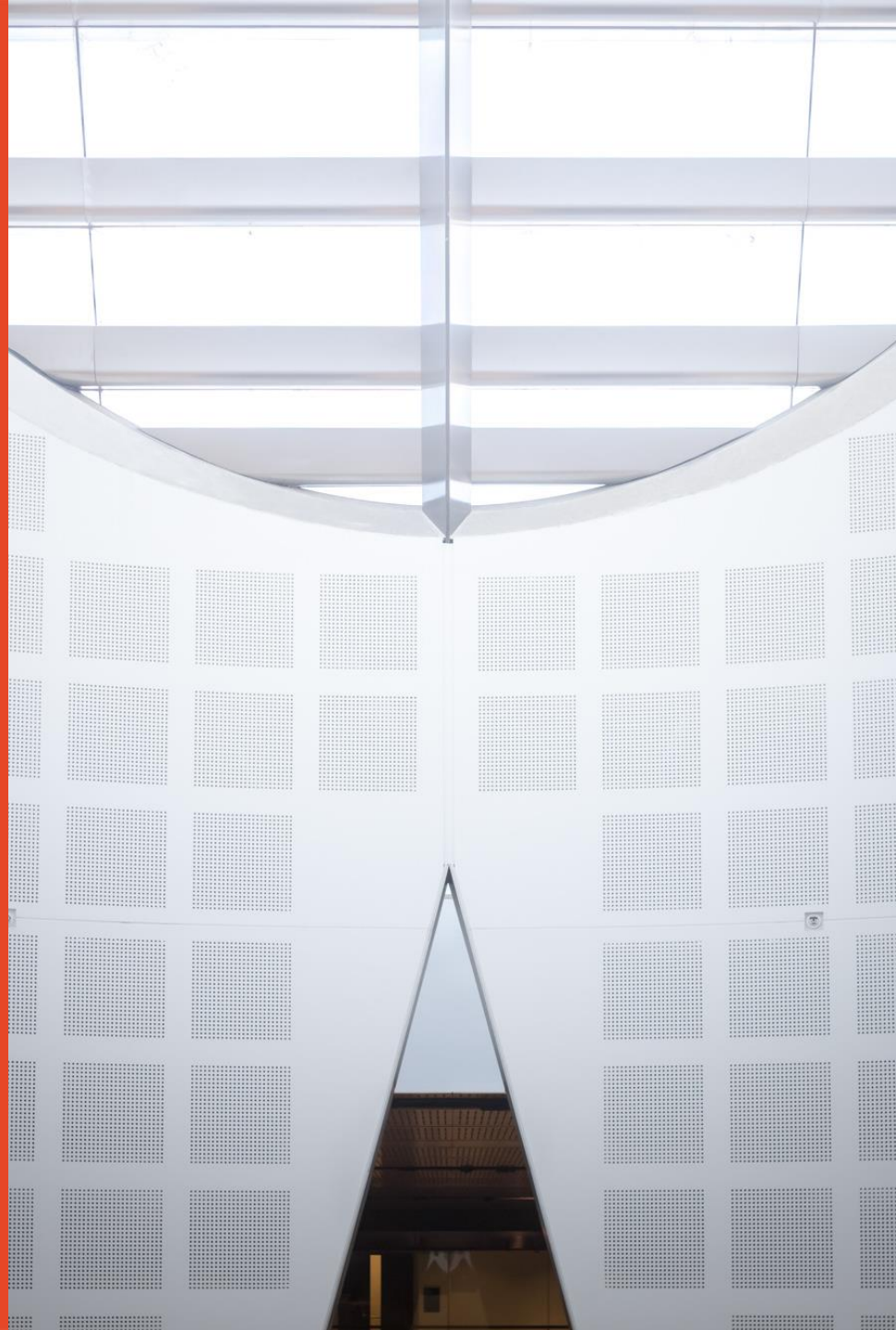
INFO1111: Computing 1A Professionalism

Week 6: Systems Thinking Problem Solving

Systems



THE UNIVERSITY OF
SYDNEY



Turing Patterns

$$\partial_t \mathbf{a} = \mathbf{D} \nabla^2 \mathbf{a} + \mathbf{R}(\mathbf{a})$$



https://en.wikipedia.org/wiki/Reaction%E2%80%93diffusion_system

Emergent Behaviours / Patterns



https://www.youtube.com/watch?v=0dskCpuxqtl&ab_channel=CGTN



<https://en.wikipedia.org/wiki/Emergence>

Systems

- Can we predict how a system will behave based on the behaviour of its parts?
 - What if each of those parts is very simple?
 - Consider basic rules for ant behaviour
 - Chaos Theory / Fractals / ...
 - Turing's Reaction-Diffusion Model
 - *Might* explain how a simple chemical interaction can lead to complex patterns in biology
- Can we determine what simple components, when combined, will give us the desired behaviour?



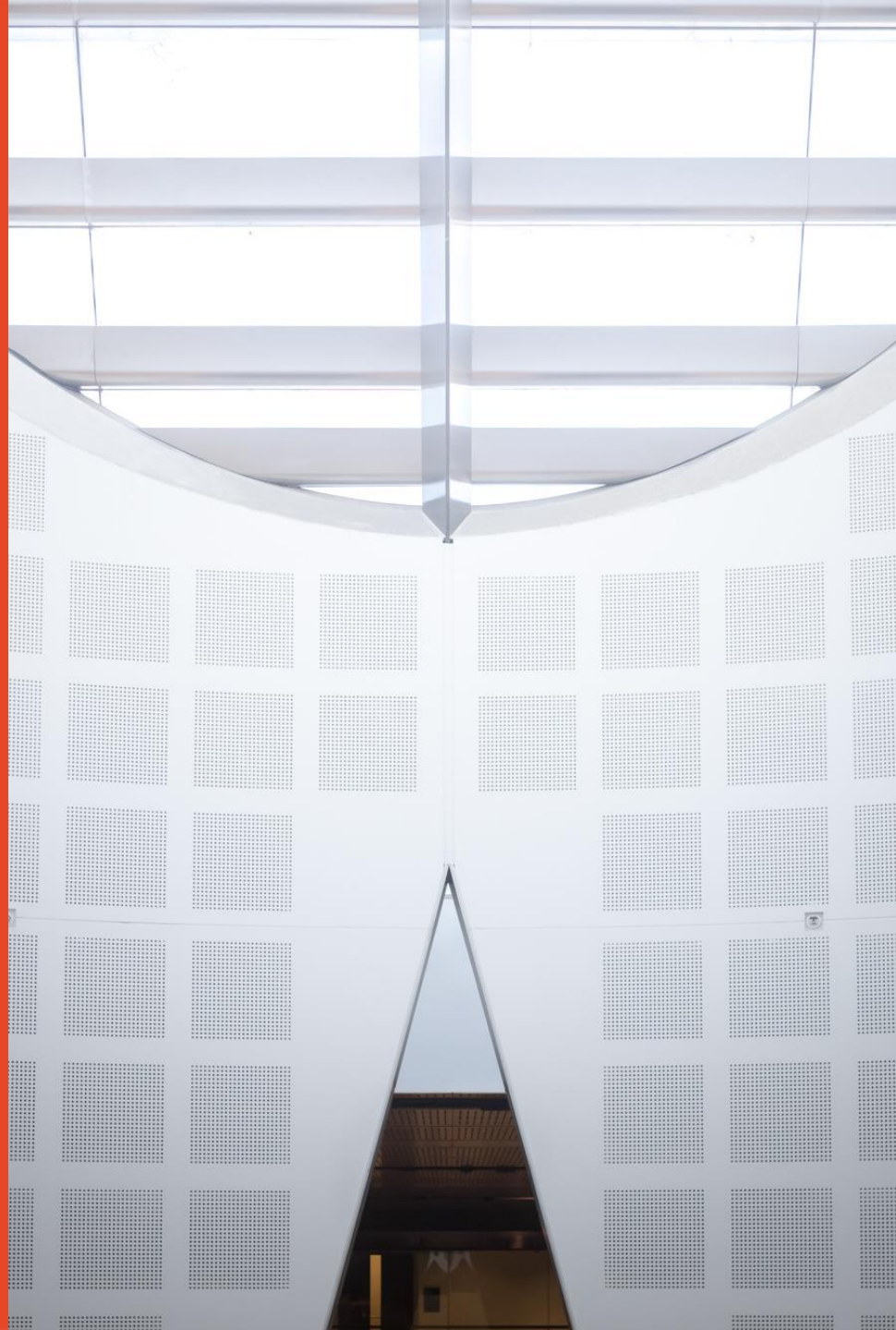
INFO1111: Computing 1A Professionalism

Week 6: Systems Thinking Problem Solving

Systems Thinking



THE UNIVERSITY OF
SYDNEY



What is “Systems Thinking”

- Broad level: The connection between solutions, systems that implement them and the society they operate in.
 - How does the work IT Professionals do fit in with the rest of the world?
- More focussed: Components of a system, their interactions and interrelationships can be analysed individually to see how they *influence the functioning of the whole system*.
 - How do our solutions influence the systems they operate in?
- ... it’s a way to view interactions in the wider world, a “way of thinking” or a “philosophy” for some.

Why is this different to other forms of thinking?



systems thinking vs



systems thinking vs **design thinking**

systems thinking vs **linear thinking**

systems thinking vs **critical thinking**

systems thinking vs **reductionism**

systems thinking vs **process thinking**

systems thinking vs **traditional thinking**

systems thinking vs **lean**

systems thinking vs **strategic thinking**

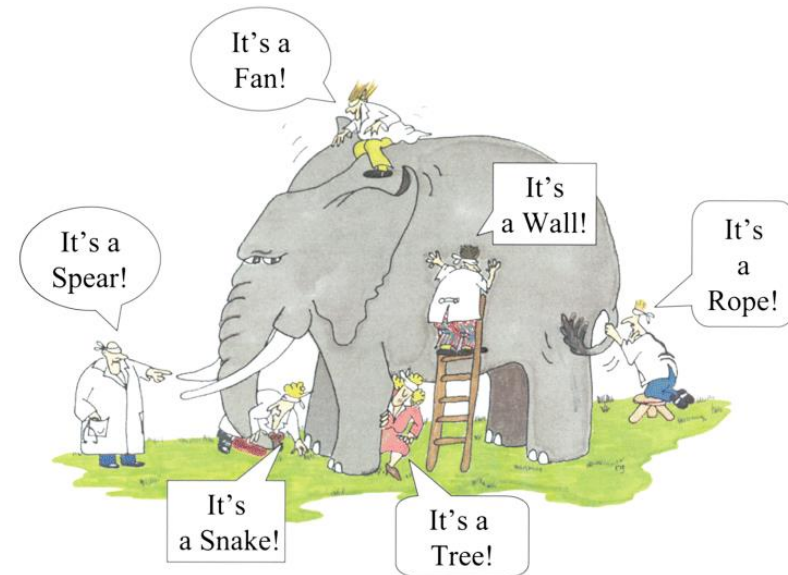
systems thinking vs **systems engineering**

systems thinking vs **peace thinking**

Report inappropriate predictions

Why is this different to other forms of thinking?

- Contrast to “traditional” systems analysis where a problem is broken into parts and studied individually
- In IT we need both ...
 - Computers and computational solutions currently focussed on “parts” and “interfaces”
 - But this is a limited view
 - Systems are highly dependent on “structure” or “relationships” among parts – how the parts are connected and this needs focus



See <https://www.patheos.com/blogs/driventoabstraction/2018/07/blind-men-elephant-folklore-knowledge/>

Why is this different to other forms of thinking?

- Systems are **complex** and **dynamic**
 - Feedback loops (balancing, or reinforcing)
 - Causality (not always clear)
 - Emergent behaviour (e.g. adapting products for tasks that designers never intended – safety concerns)
 - Behaviour over time can vary
 - And they can be unpredictable

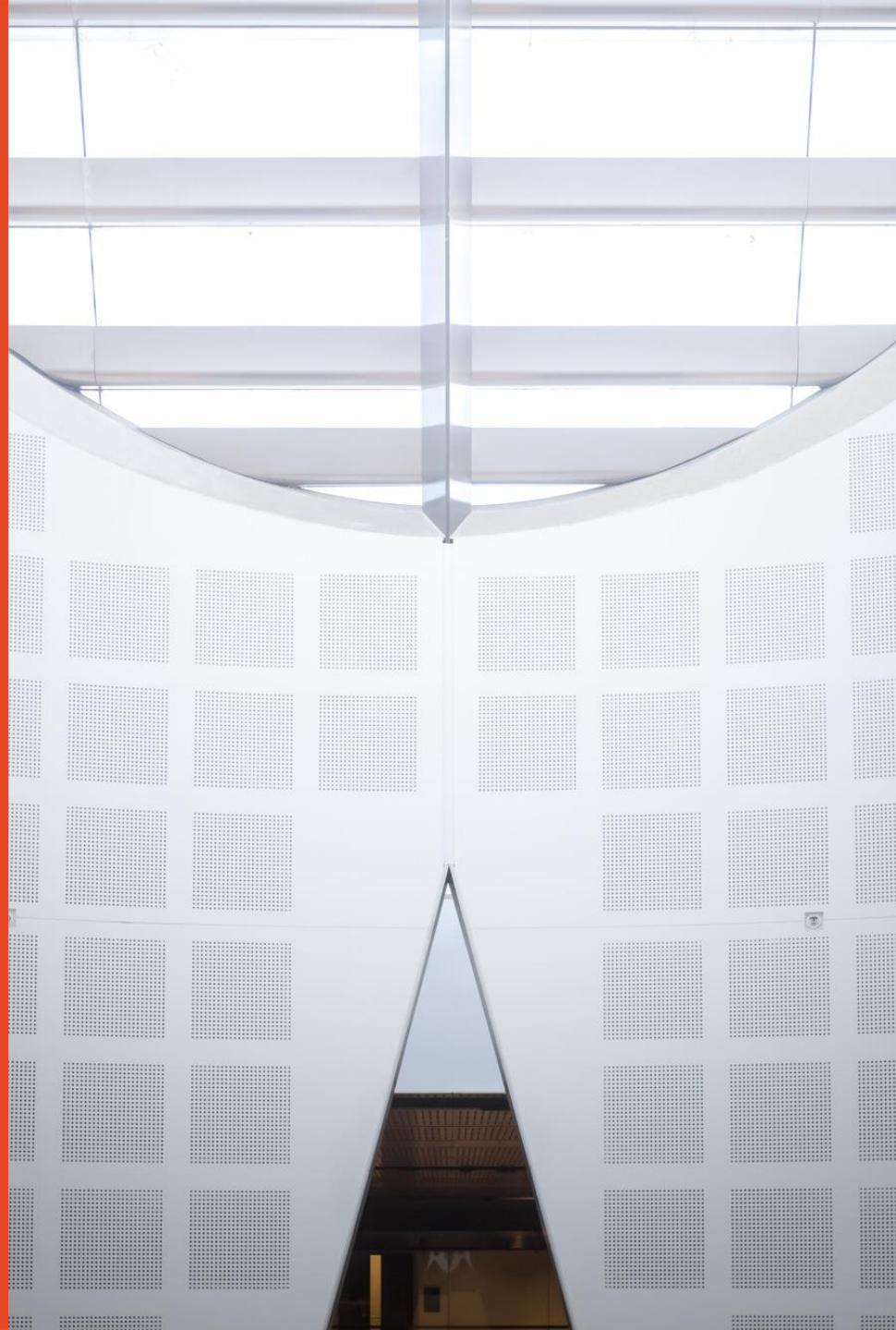
INFO1111: Computing 1A Professionalism

Week 6: Systems Thinking Problem Solving

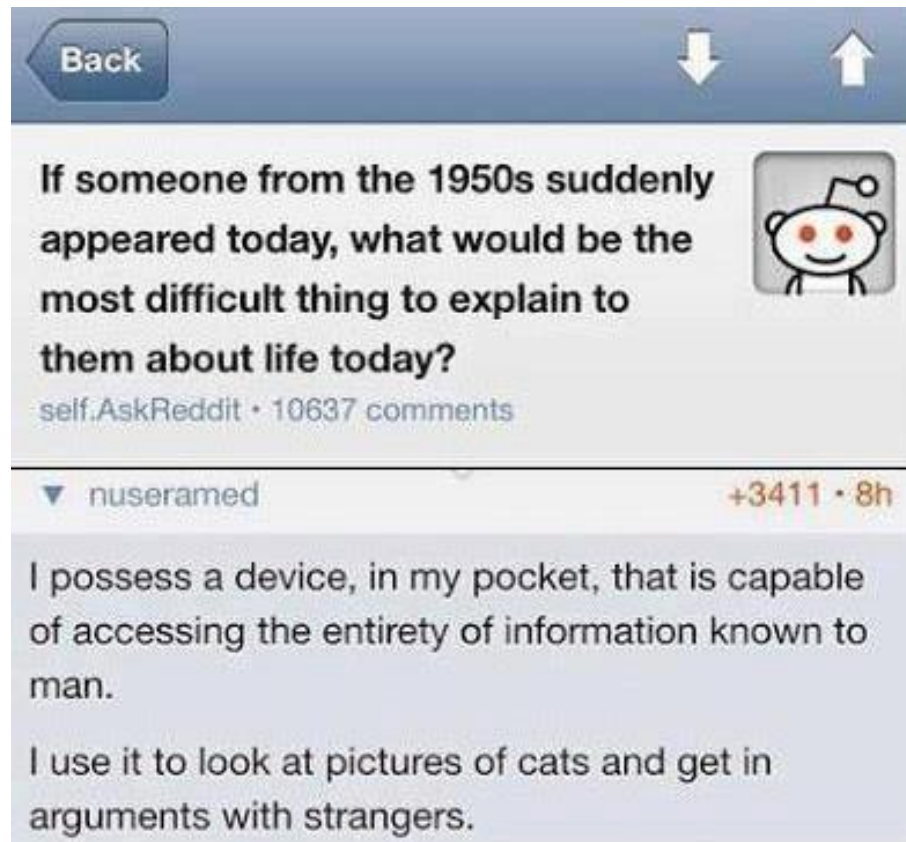
Systems Thinking (part 2)



THE UNIVERSITY OF
SYDNEY



Why is this important?



- Unintended consequences
- Complex systems fail

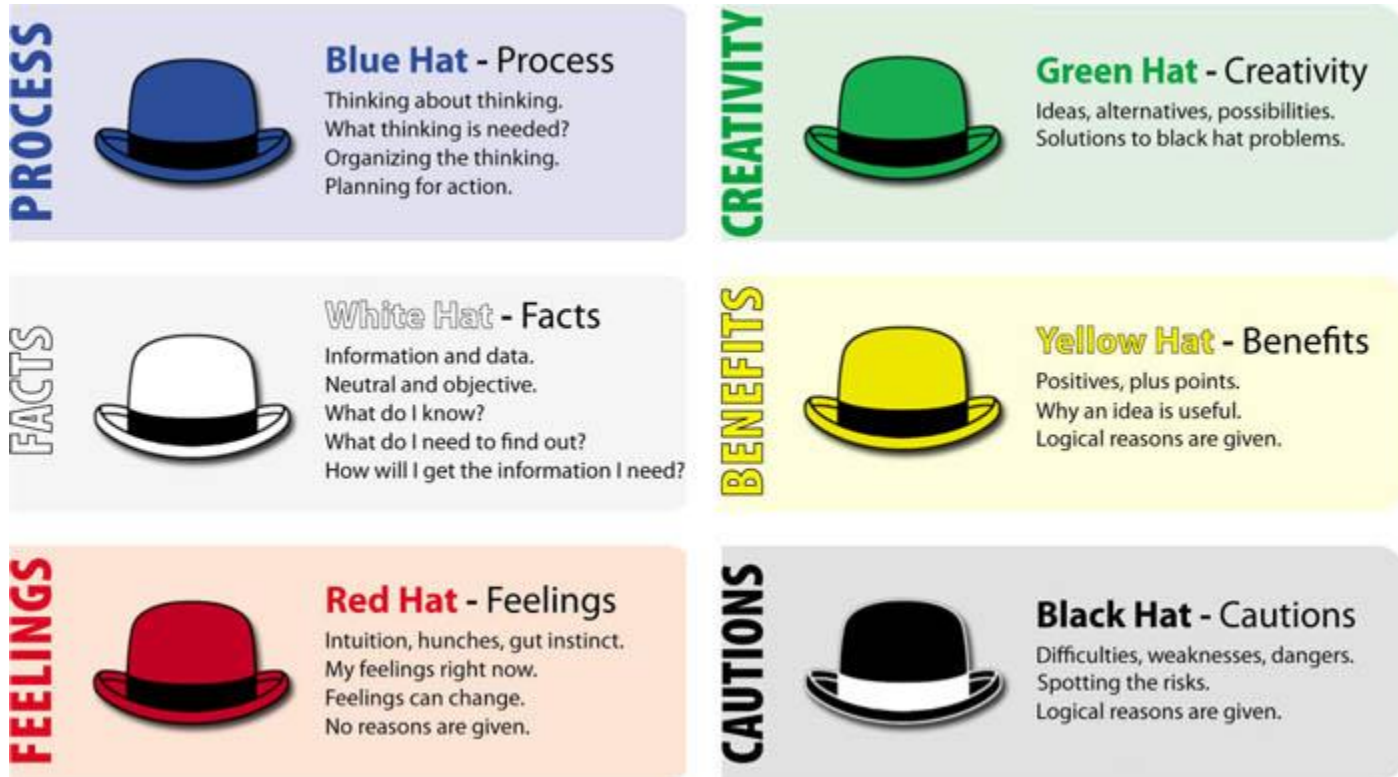
Why is this important?

- Tragedy of Commons
- Borneo sprayed with DDT to kill mosquitoes
 - Reduced malaria by killing mosquitoes
 - ... but roofs started falling in
- Therac-25 (an example for almost everything ...)
 - “The equipment control task did not properly synchronize with the operator interface task, so that race conditions occurred if the operator changed the setup too quickly. This was missed during testing, since it took some practice before operators were able to work quickly enough to trigger this failure mode.” <https://en.wikipedia.org/wiki/Therac-25>
 - Solution exists in bigger system
- “Failure is almost never obvious until you’re looking in the rearview mirror”
 - See <https://www.oreilly.com/ideas/the-critical-role-of-systems-thinking-in-software-development>

Why is this important?

- System thinking: identify the interdependencies between elements of a system as feedback cycles and delays
 - What's perceived as an isolated problem can be part of an interconnected network of other issues
 - Identify positive and negative feedback cycles affecting a problem
 - Feedback cycles could take a long time
 - Problems could be the result of previously unrecognized feedback cycles
 - What the system is susceptible to can be leveraged to create change
 - What seems to worsen the situation may, in fact, lead to identify other solutions
 - <http://www.systemsthinker.com/interests/systemsthinking/>

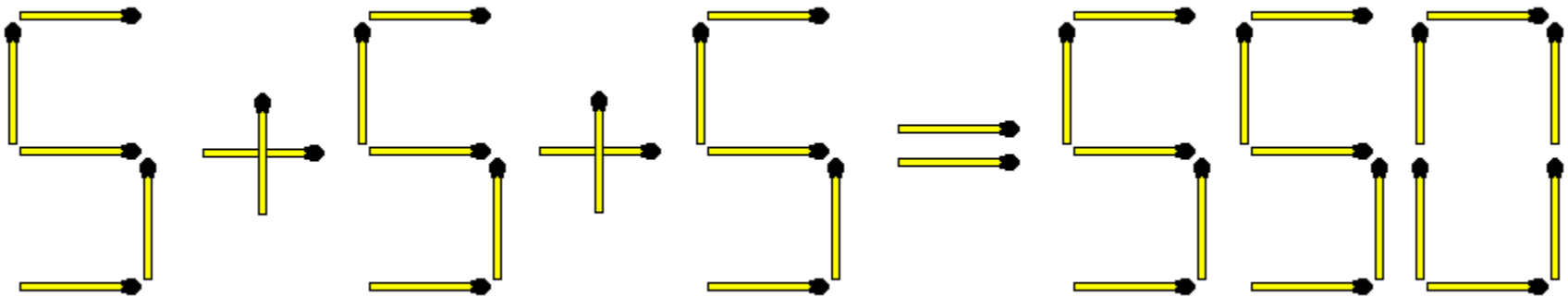
Problem solving



<https://jliongue.wordpress.com/2015/09/17/week-five-reading-six-thinking-hats/>

Problem Solving

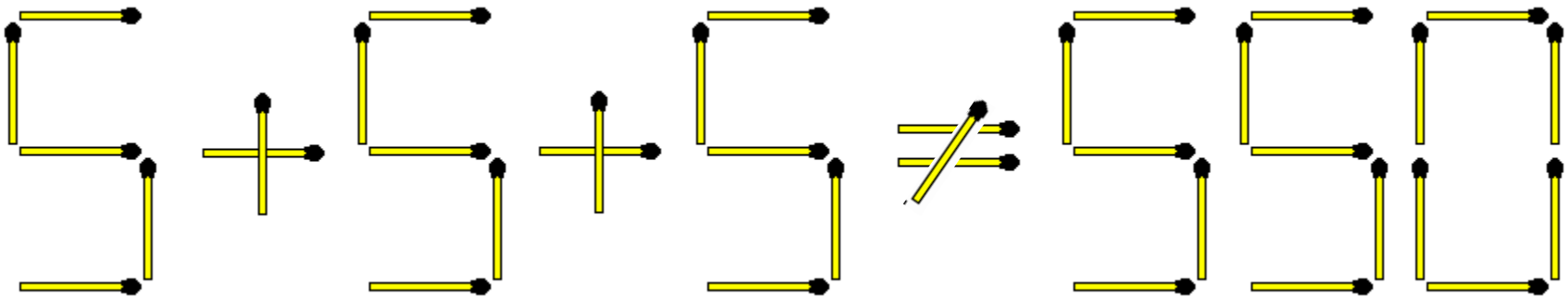
- How would you go about solving these problems ...



Add only 1 matchstick to make the equation true.

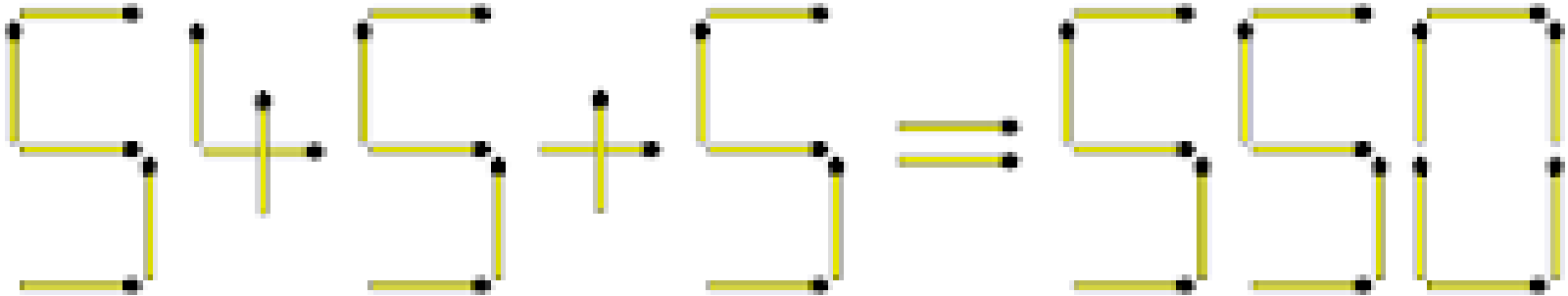
Problem Solving

- How would you go about solving these problems ...



Now do it so that you still have “=”?

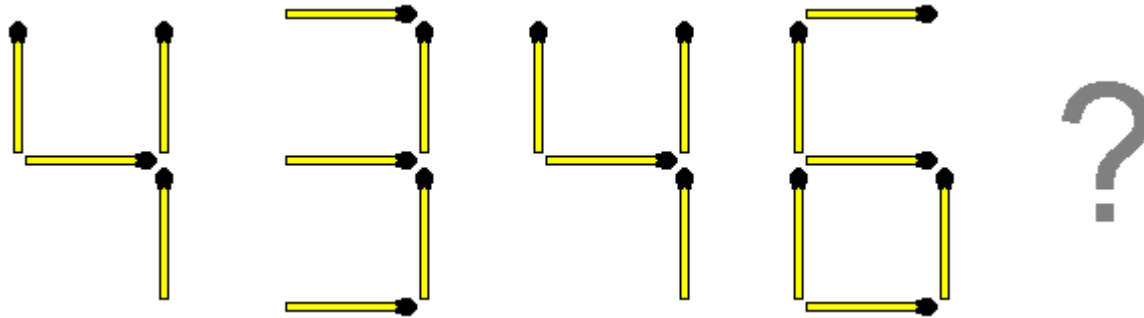
Problem Solving



- The problem here is clear but the solution is tricky

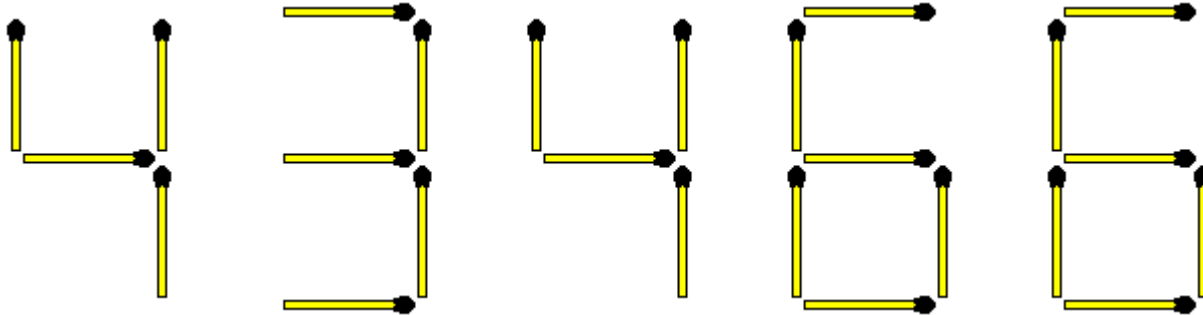
Problem Solving

- Sometime it is the problem that is tricky ...



FIND THE NEXT NUMBER... PLEASE

Problem Solving



- The number of letters in the different words of the question represent the numbers.

FIND = 4

THE = 3

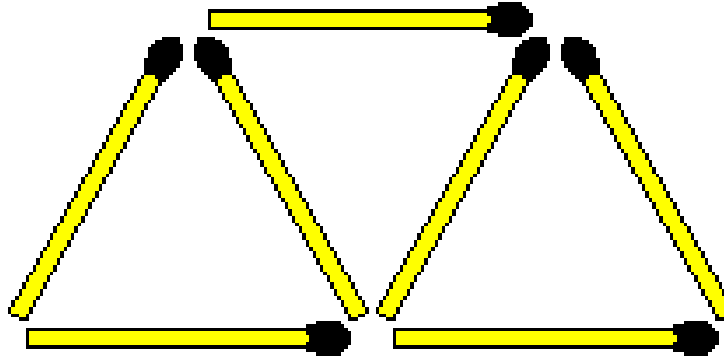
NEXT = 4

NUMBER = 6

PLEASE = 6

Problem Solving

- Try this one ...

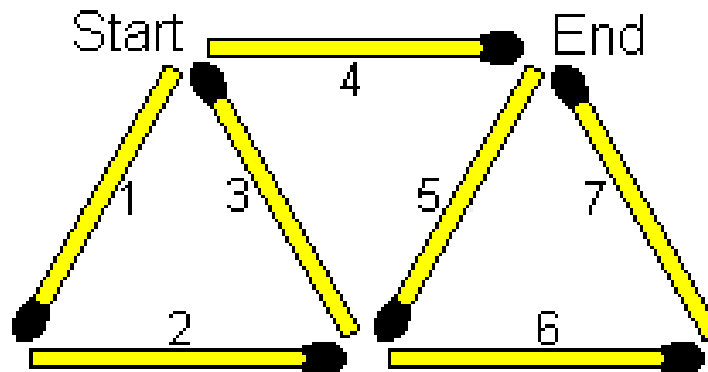


7 matchsticks are used to form a pathway as shown above. The challenge is to start at a point, move in the direction of the matchstick heads and visit each of the 7 matchsticks only once.

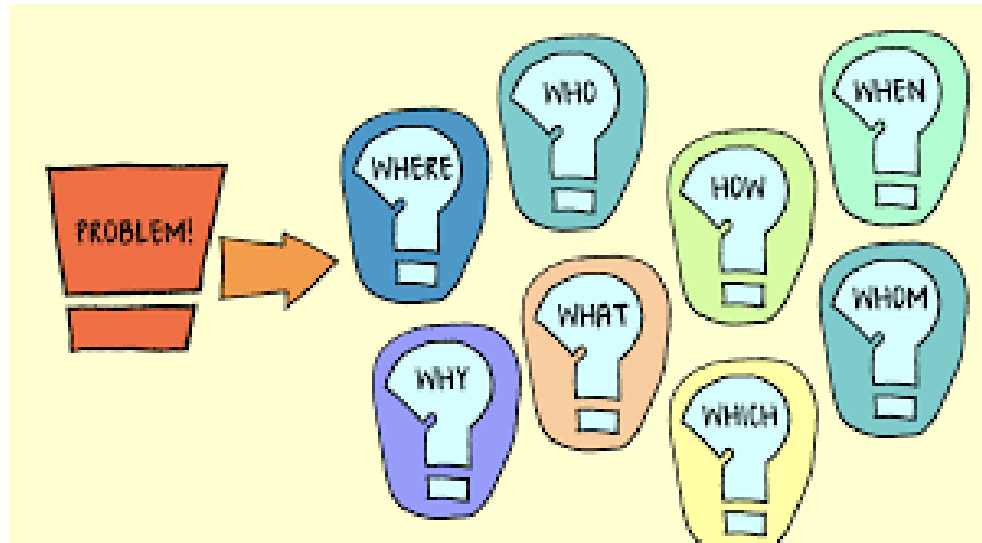
A solution is not possible in the existing figure above, but by changing the direction of only 2 matchsticks it can be achieved. How would you do this?

Problem Solving

- First ... how did you approach this problem?
 - Trial and error?
 - A more systematic approach?
 - Do you have a memory of having solved a problem like this before?



Problem solving

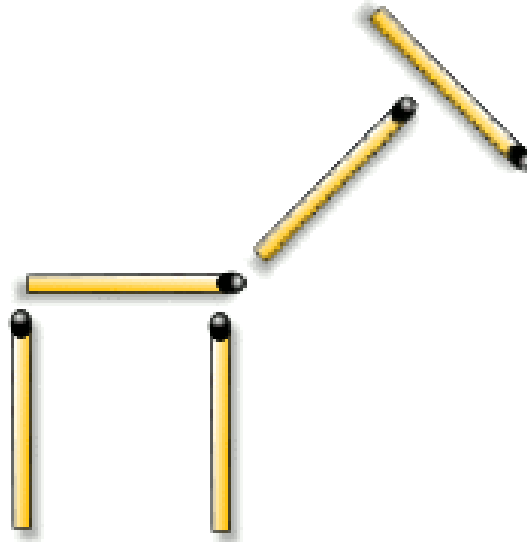


Source:medium.com

What sort of problems do you think Apple had ?



Problem Solving



Move one matchstick to turn the donkey
(the donkey must remain the same size and shape)

Problem Solving



- How was this problem different?

Understanding the Problem

- The last two puzzles emphasise the point that before solving a problem, you need to understand what the problem is!
- In a professional IT environment this is often:
 - Problem specification
 - Requirement specification
 - Understanding constraints
 - etc

It is not always what you think ...

Once upon a time, several floors were added to the top of a hotel. With that expansion came new elevators.

With the new elevators came new complaints. “The elevators are too slow. We have to wait too long. I’m annoyed.”

The hotel management was aghast. They consulted architects. They consulted engineers. What to do?

One possibility was to install faster elevators...at massive expense. Another possibility was to add another elevator shaft with more elevators...at massive expense.

The expense of improvements required to speed up the elevators would essentially eliminate the benefit of having the extra floors! It was a disaster.

It is not always what you think ...

Someone tried an experiment. They mounted full-length mirrors next to the elevators.

The complaints stopped. Why?

It would appear that the real problem was not, “The elevators are too slow.” Instead, the real problem was, “Guests get annoyed while waiting for the elevator.”

- <http://www.realitycheck.focusonclarity.com/archives/1372>



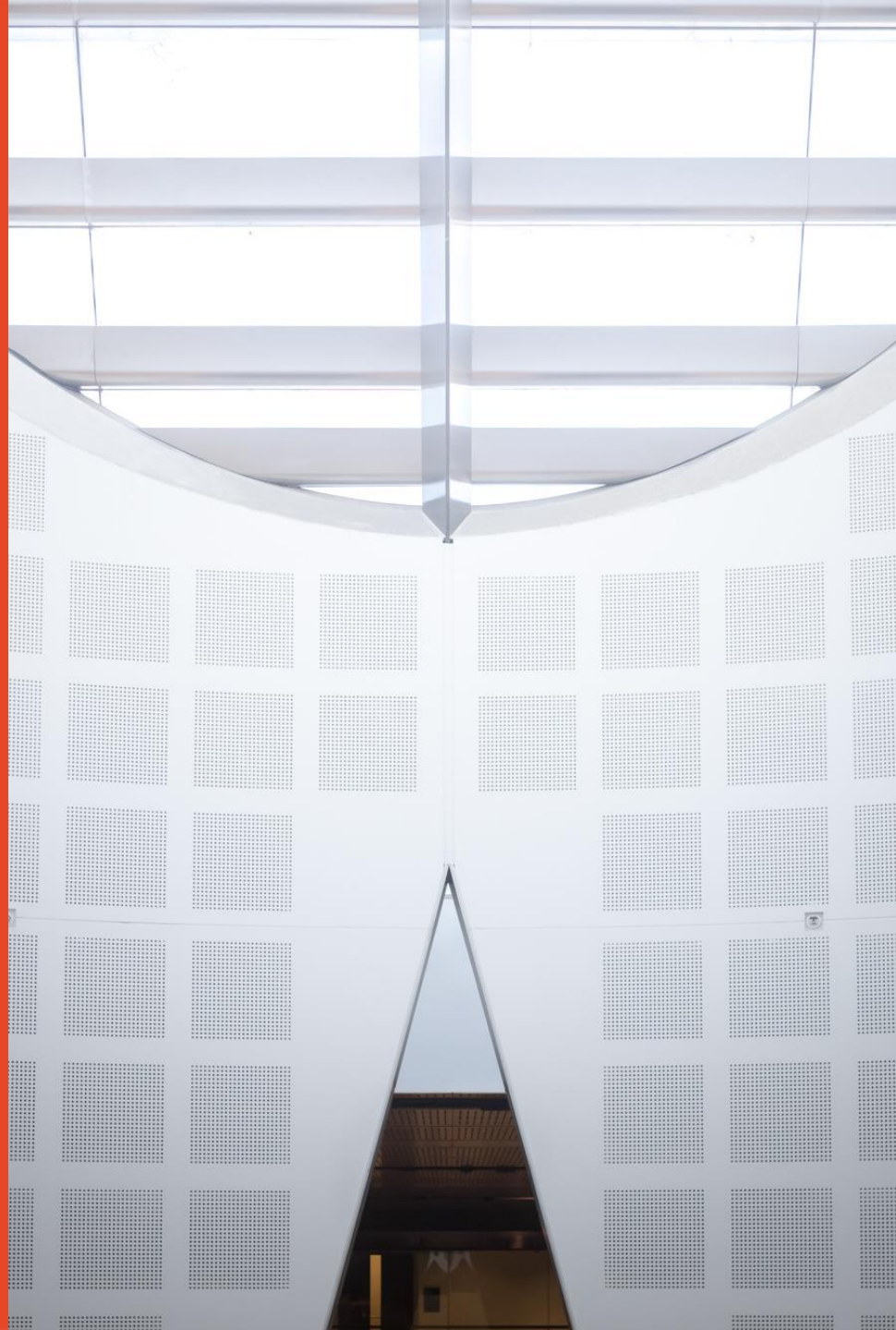
INFO1111: Computing 1A Professionalism

Week 6: Systems Thinking Problem Solving

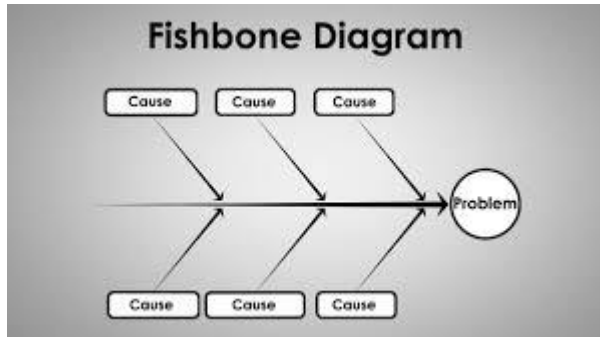
Problem Solving (part 2)



THE UNIVERSITY OF
SYDNEY



Problem solving approaches



Source: onvergencetraining.com



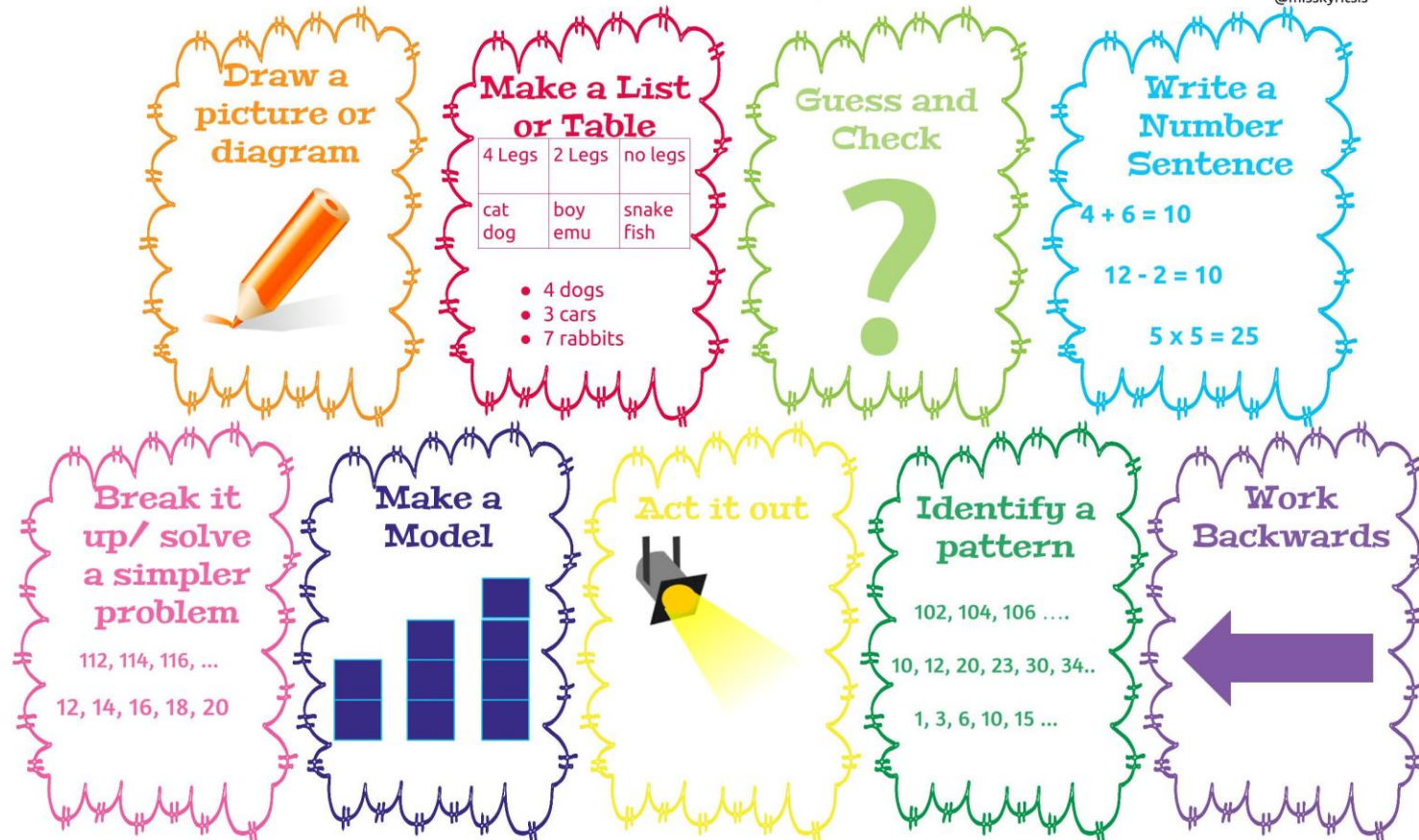
Source: youtube.com

Can you think of others?

Problem Solving approaches

Problem Solving Strategies

@misskyritsis



Source:elenikyritsis.com

Understanding the problem

- Bad requirements may be:
 - incorrect – incomplete – vague – conflicting – missing – solution focused
- Examples
 - The system shall have good usability
 - The system shall respond very fast
 - The system shall respond in less than 2 seconds
 - The system shall be error-free

Salaries for Business Analysts / problem solvers

- From <https://www.payscale.com/research/AU>
- Software Tester - \$62,218
- Programmer - \$69,164
- Software Engineer - \$74,963
- **Business Analyst - \$81,629**
- IT Project Manager - \$103,133

- From <http://www.glassdoor.com.au>
- Programmer - \$69,340
- Software Engineer - \$84k
- **Business Analyst - \$84k**
- Senior Software Engineer - \$104k
- IT Project Manager - \$128k

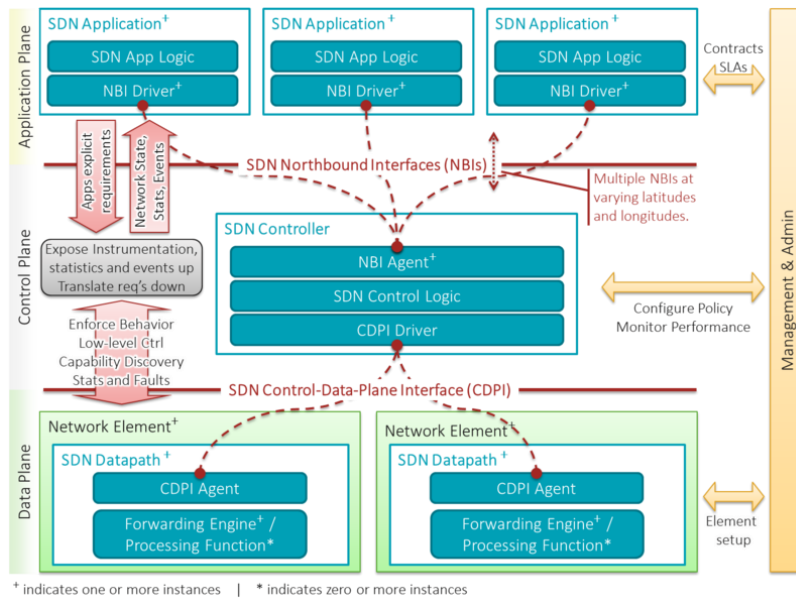
Problem solving...

- Problem solving is: Challenging; Rewarding; Creative
- Approaches: identifying possible alternatives
 - Brainstorming: forming; STORMING; norming; performing
 - Word association; mind mapping; word banks; visual association; ask what if; exaggeration; think prepositionally; ask why; role storming; super powers; SWOT analysis; question assumptions
 - Decomposing problems
 - Break the problem into parts (divide and conquer). But how?
 - Functional? Procedural? Domain? Role?
 - Minimise coupling – the dependencies between components
 - Maximise cohesion – the focus of each component
- Can you learn to be creative?
 - Activity: (30 seconds) How many different things can you think of to do with 50 paper clips?

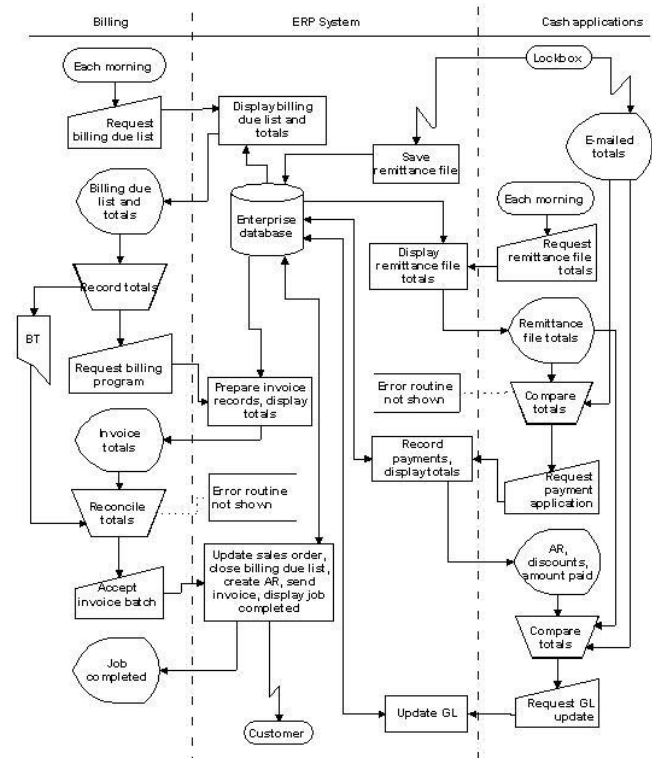
“Levels and languages” for software solutions?

– System-level: Architecture

– Unit level: Algorithms



<https://commons.wikimedia.org/wiki/File:SDN-architecture-overview-transparent.png>



https://commons.wikimedia.org/wiki/File:Stockbridge_system_flowchart_example.jpg

INFO1111: Computing 1A Professionalism

Week 6:
Systems Thinking
Problem Solving

Wrap-up



THE UNIVERSITY OF
SYDNEY

