AGILE DEVELOPMENT

BRINGING ESTIMATION TO LIFE

ESTIMATION

- Challenges
  - How do you estimate when you don't know the exact nature of the task?
  - How do you estimate for something you've never done before?
  - How do you estimate for a team?
  - How can you have confidence in your estimates?
  - How do you improve your ability to estimate?

SOLVING PROBLEMS

- Software engineers and data analysts constantly draw upon their problem solving skills

  - Whilst working on a problem you won't always know what <u>challenges</u> you may encounter

  - To solve a complex problem you need to <u>break it down</u> into smaller pieces

  - Past experiences will help you to solve "similar" problems using the <u>skills</u> you've acquired

  - You will get better with practice but even better if you take the time to <u>review</u> + reflect
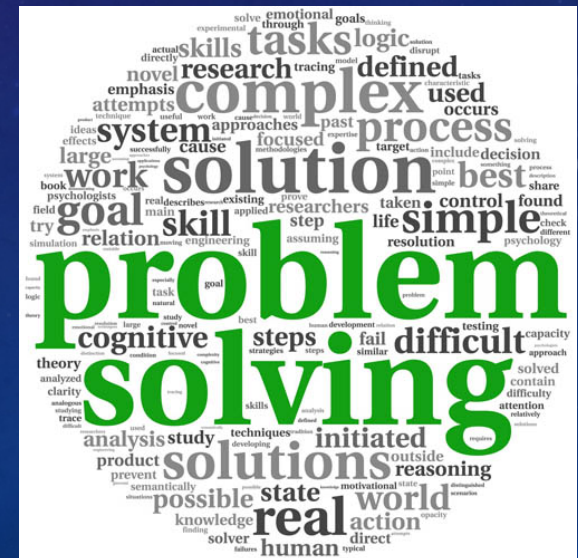
THE WORLD'S MOST FAMOUS PUZZLE

- Invented in 1974 by Ernő Rubik and originally called the "Magic Cube"

- Debuted at the toy fairs in London, Paris, Nuremberg and New York in Jan + Feb 1980

- 350 million sold worldwide by January 2009

- There are <u>many</u> permutations - 43,252,003,274,489,856,000 to be precise!

ANALOGY WITH AGILE

- Estimation challenge

  - How many <u>moves</u> will it take me to solve a Rubik's cube?

  - Bear in mind that every "solve" will be slightly different…

- Possible approaches

  1. Just estimate the <u>total</u> number of moves

  2. Split the task up, estimate the number of moves for each <u>part</u> then add them up

METHOD / FRAMEWORK - CFOP

1. Cross - Four "edges"

2. First Two layers (F2L)
   - Four "corner + edge" pairs

3. Orientation of the Last Layer (OLL)

4. Permutation of the Last Layer (PLL)

ACTUALS VS ESTIMATES

- Cross - Typically ~6 moves

- F2L - Typically ~30 moves

- OLL - Typically ~11 moves but can be as high as 14 moves

- PLL - Typically ~13 moves but can be as high as 21 moves

- TOTAL ~60 moves

RELATIONSHIP TO AGILE

- This is a relatively simple task / problem but there are similarities with agile methodologies
  - Breaking larger tasks into smaller tasks even though you won't know exactly what challenges will be encountered
  - Your ability to handle different scenarios improves with experience. Figuring out your own solutions is great but you can often learn from others
  - Common patterns / scenarios become familiar and the ability to identify / apply an appropriate solution improves with practice
    - You'll spend less time considering inappropriate solutions and quickly settle upon an suitable solution
    - You'll become more accurate with practice, reducing mistakes and minimising the amount of rework / unnecessary effort
  - When you start out, estimates may be little more than guesswork but experience and learning from others will improve your estimation skills
  - Estimating on behalf of other people (or teams) is likely to pose a challenge until you know their skill level and their delivery record
  - Hard figures can be recorded and analysed to see how the outcomes vary from your expectations (e.g. actuals VS estimates)
  - As skill levels increase, estimates will change accordingly (e.g. tasks require less effort) and estimates become more accurate
  - However… there will always be some variation even for similar tasks. How can we be confident about delivering by the end of the sprint?
    - Ensure that you include some contingency. e.g. Consider allocating 85% of the available resources to sprint related tasks and 15% for Kanban tasks