PROBLEM SOLVING + BASIC SKILLS

TODO TODAY

- Basic skills
- Problem solving

BASIC SKILLS

What is taught on real courses:

- Programing language
  - Control flow, methods/function
  - OORPG, OOP, ITS1, ITS2, Programmering, ..
- Data structures
  - List, queue, stack, tree
  - DOA, Algoritmer og Datastrukure
- Software architecture & design
  - SWD, ITS3, Software Architecture,

LEARN YOU EDITOR?

- Shortcuts (VS Code, VS 2019, Rider)
- Views
- Plugins / extensions

C / C++ / C#

- Visual Studio does everything
  - But what actually happens?

C / C++ / C#

- Visual Studio does everything
  - But what actually happens?
- C & C++ The compiler:
  - E.g. GCC

C / C++ / C#

- Visual Studio does everything
  - But what actually happens?
- C & C++ The compiler:
  - E.g. `GCC`
- For C# the compiler is called:
  - .Net Core: `dotnet build`
  - .Net: `csc.exe`

C / C++ / C#

- Visual Studio does everything
  - But what actually happens?
- C & C++ The compiler:
  - E.g. `GCC`
- For C# the compiler is called:
  - .Net Core: `dotnet build`
  - .Net: `csc.exe`

* Many errors are reported by the compiler

JAVASCRIPT

- JavaScript files can be 'executed' directly in browser.

JAVASCRIPT

- JavaScript files can be 'executed' directly in browser.
- TypeScript (Superset of JavaScript)
  - Must be transpiled to JavaScript
  - Requires a tool chain tools e.g. Gulp

JAVASCRIPT

- JavaScript files can be 'executed' directly in browser.
- TypeScript (Superset of JavaScript)
  - Must be transpiled to JavaScript
  - Requires a tool chain tools e.g. Gulp

```
gulp.task('serve', () => {
    connect.server({
      root: root, port: port,
      host: '0.0.0.0', livereload: true
    })

    gulp.watch(['*.html', '*.md'], gulp.series('reload'))
    ...
```

TOOLCHAINS

- Do you use any other tools?
  - Cross compilation, linting, obfuscating, etc.?
  - Are errors coming from these?

WHAT IS GOOD CODE?

- Correct indention
- Readability
- High Cohesion
- Low Coupling

```
if(a < EARLIEST_HOUR && b < EARLIEST_HOUR
&& c < EARLIEST_HOUR)
a = EARLIEST_HOUR; b = EARLIEST_HOUR; c = EARLIEST_HOUR;
```

```
public class TennisGame3 : ITennisGame {
  private int p2; private int p1;
  private string p1N; private string p2N;

  public TennisGame3(string player1Name, string player2Name)
  {
    this.p1N = player1Name;
    this.p2N = player2Name;
  }

  public string GetScore() {
    string s;
    if ((p1 < 4 && p2 < 4) && (p1 + p2 < 6)) {
      string[] p = { "Love", "Fifteen", "Thirty", "Forty" };
      s = p[p1];
```

WHEN IS IT GOOD ENOUGH?

- Does it solve the problem?
  - Look at problem statement / exercise desciption

WHEN IS IT GOOD ENOUGH?

- Does it solve the problem?
  - Look at problem statement / exercise desciption
- Any known bugs?
  - Small cosmetic errors?
  - Big functional errors?

WHEN IS IT GOOD ENOUGH?

- Does it solve the problem?
  - Look at problem statement / exercise desciption
- Any known bugs?
  - Small cosmetic errors?
  - Big functional errors?
- Do you know how to solve these?

PROBLEM SOLVING

- *Always* have a plan
- Restate problem
- Divide the problem
- Start with what you know
- Reduce the problem

HALF OF A SQUARE

Write a program that given a number lets say 5, prints half of a 5*5 square.

```
#####
####
###
##
#
```

SOLVING THE PROBLEM

- You properly know how to print a full square given the number 5

```
#####
#####
#####
#####
#####
```

SOLVING THE PROBLEM

- You properly know how to print a full square given the number 5

```
#####
#####
#####
#####
#####
```

Or at least

```
#####
```

COUNTING DOWN

Can you write an `expression` in the loop such that it prints the numbers 5 through 1?

```
for (int row = 1; row <= 5; row++) {
   Console.WriteLine(`expression`);
}
```

COUNTING DOWN

Can you write an `expression` in the loop such that it prints the numbers 5 through 1?

```
for (int row = 1; row <= 5; row++) {
   Console.WriteLine(`expression`);
}
```

Maybe `'row * -1'`

CONTING DOWN (CONT.)

Row	Desired	'Row * -1'	Difference
1	5	-1	6
2	4	-2	6
3	3	-3	6
4	2	-4	6
5	1	-5	6

CONTING DOWN (CONT.)

Row	Desired	'Row * -1'	Difference
1	5	-1	6
2	4	-2	6
3	3	-3	6
4	2	-4	6
5	1	-5	6

- Formula must be $(\text{Row} * -1) + 6$
  - or just $6 - \text{Row}$

A FINAL SOLUTION

```
for (int row = 1; row <= 5; row++) {
    for (int j = 1; j <= 6-row; j++) {
        Console.Write("#");
    }
    Console.Write("\n");
}
```

LETS SLICE AN ELEPHANT

USER STORIES VS USE CASE

"User Stories are centered on the result and the benefit of the thing you're describing, whereas Use Cases can be more granular, and describe how your system will act"

- Both has a role, goal, and acceptance
- User story contain fewer details
- Use cases have all details up front

USER STORIES

We will use this definition:

User story: vertical, testable, user valued.

USER STORIES

- Vertical:
  - Cut accross multiple layers in the architecture
- Testable:
  - Should be testable either by code or manually
- User valued:
  - Should bring value to end user.

USER STORIES EXAMPLES

Is removing:

```
Console.WriteLine("Hello World")
```

a user good story?

USER STORIES EXAMPLES

Is removing:

```
Console.WriteLine("Hello World")
```

a user good story?

Adding a button to UI?

USER STORIES EXAMPLES

Is removing:

```
Console.WriteLine("Hello World")
```

a user good story?

Adding a button to UI?

Setting up project structure?

USER STORY SIZE

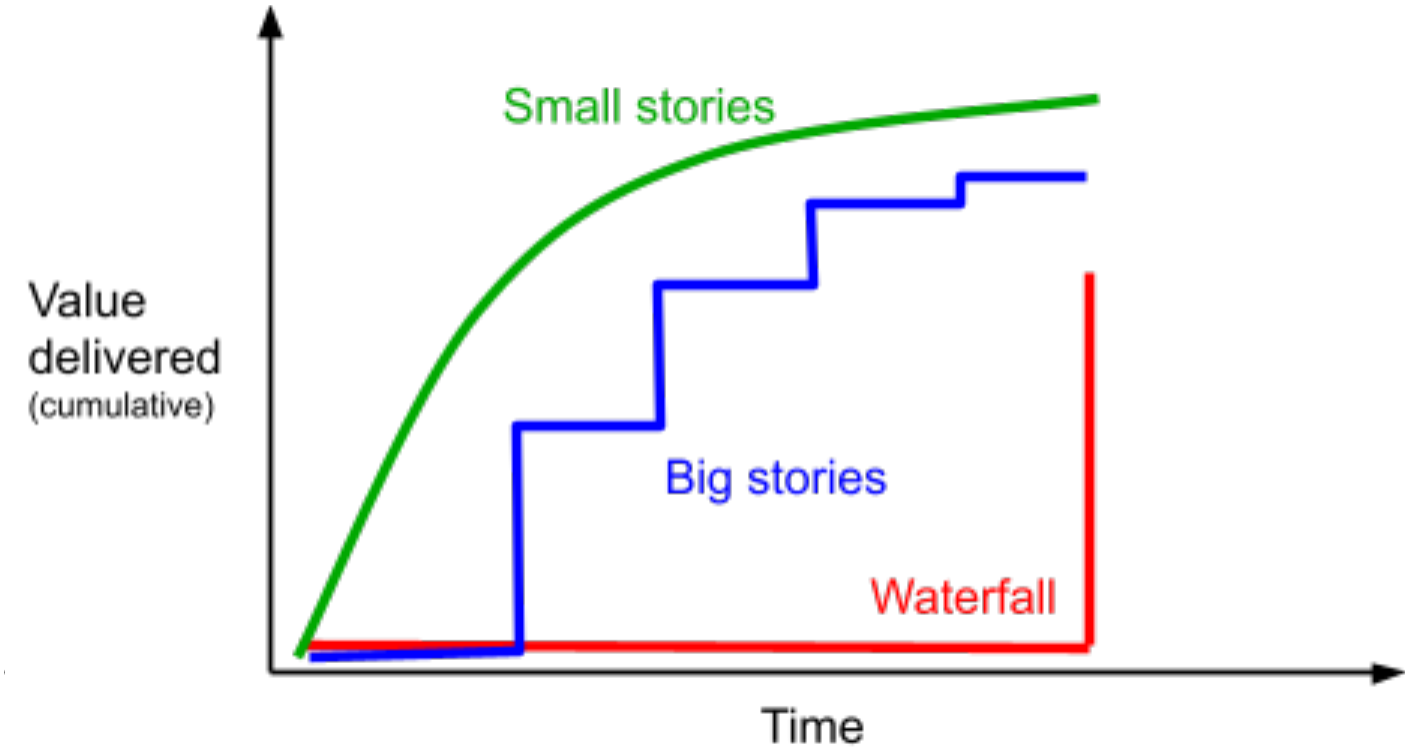
- Effort
- Complexity
- Unknowns

USER STORY - HOW BIG?

From minutes to months.

WHY SMALL STORIES?

- Problem is divided into smaller parts
- Easier to make a plan
- Easier to recognise known parts


Small stories
Value
delivered
(cumulative)
Big stories
Waterfall
Time

EXERCISE

- Divide into 2-3 persons groups
  - Try meeting new people :-)
- 15 min for breaking down problem into small user stories
- If time allows
  - Start implementing in 8 minute development sprints

PRODUCT

- **User input:**
  - How many items
  - Price per item
  - 2-letter state code
- **Output:**
  - Total price
    - Discount based on total price
    - Add state tax based on state and discounted price

Order value	Discount rate	*	State	Tax Rate
1.000	3%	*	UT	6.85%
5.000	5%	*	NV	8.00%
7.000	7%	*	TX	6.25%
10.000	10%	*	AL	4.00%
50.000	15%	*	CA	8.25%

USER STORIES

1. Input #item
2. Validate input
3. Input price per item
4. Validate price
5. Sum value
6. Present total
7. Calculate discount for
8. 1000
9. 5000
10. 7000
11. 10.000
12. 50.000
13. Input state code
14. Validate state code
15. Present error
16. Calculate tax rate for
17. UT
18. NV
19. TX
20. AL
21. CA

REFERENCE

- Think like a programmer by V. Anton Spraul
- Giphy
- https://kata-log.rocks/elephant-carpaccio