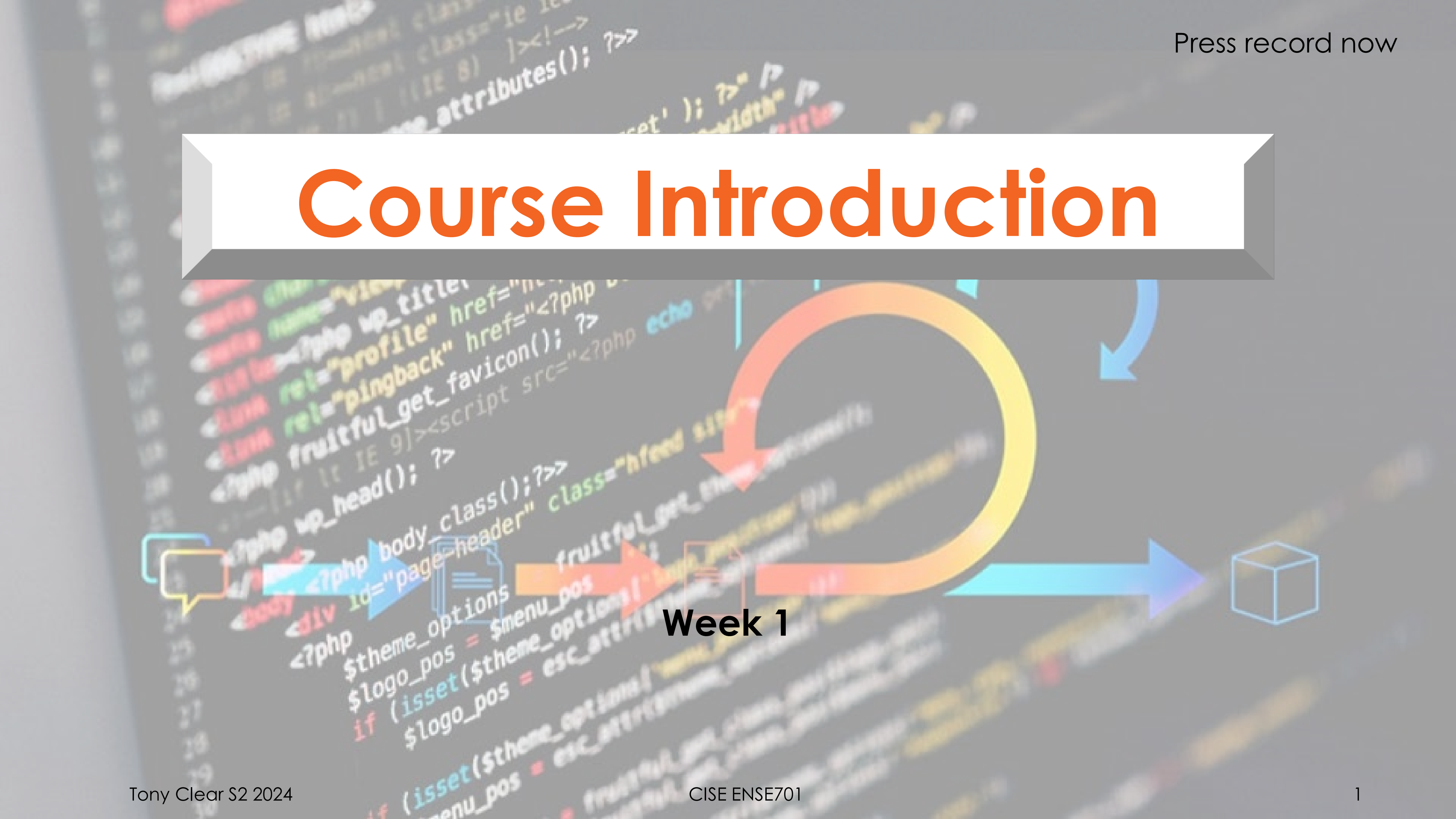
Press record now

Course Introduction



Week 1

Mihi

Hāere mai, Haere Mai, Haere Mai.

Tēnā koutou katoa.

Ko Tony Clear taku ingoa.

Nō Pōneke ahau.

Ko Maungakiekie taku maunga.

Ko Waitematā taku moana.

I te taha o taku matua, no Enniscorthy Ireland ahau.

I te taha o aku whaea, no Cork Ireland ahau.

Ko Tainui Raua Ko Ngapuhi nga iwi o nga mokopuna

Tēnā koutou, Tēnā koutou, Tēnā tatou katoa.

A bit about me

First Degrees in Latin and English Language

Stint as a high school teacher

Joined IT industry as a programmer

Worked on large government and corporate systems projects

Managed teams of developers and other functions

Joined then AIT as staff manager, gained a Master’s degree in IS and PhD in Computer and Information Sciences

Held roles of discipline group leader, associate head of school, head of school, faculty associate dean research

Co-ordinated R&D project course for many years,

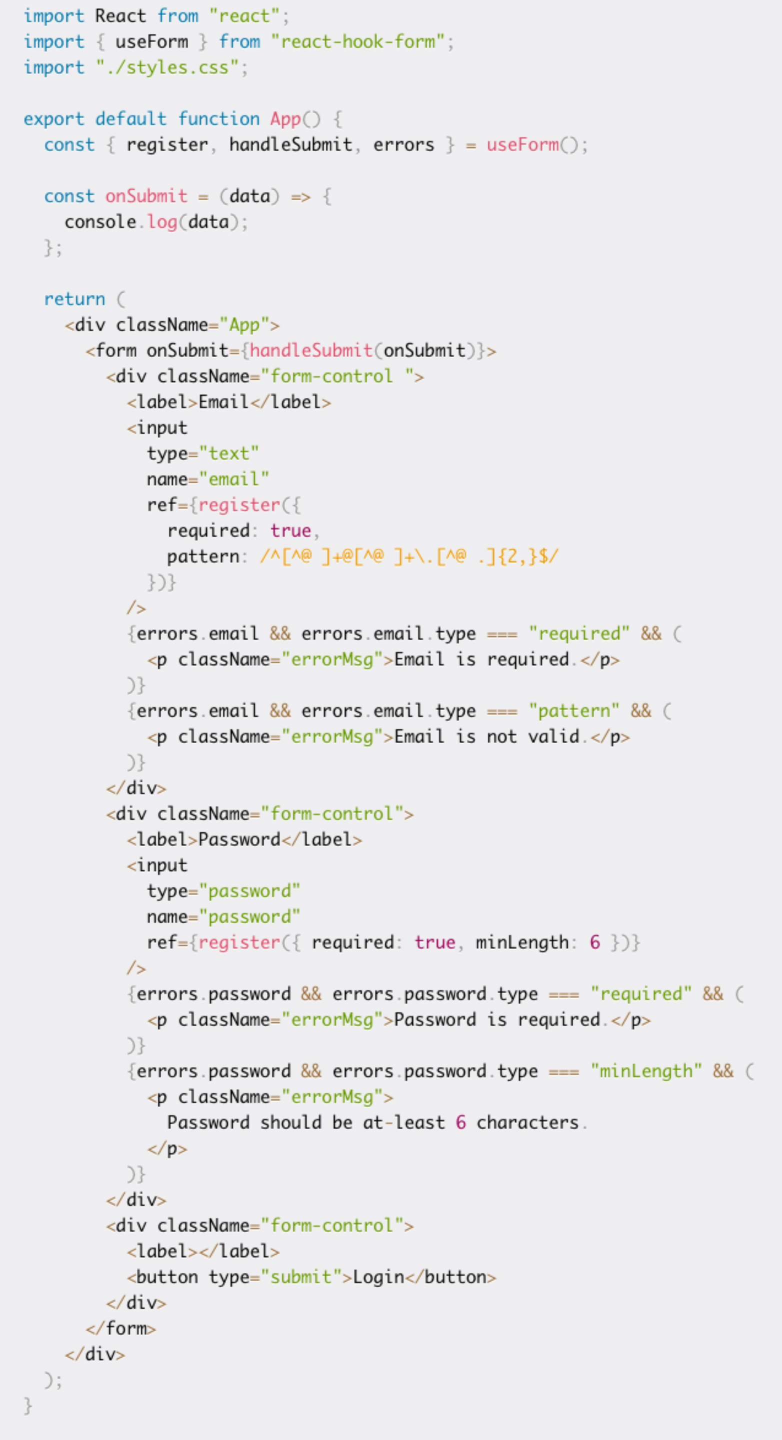
Now associate professor with teaching, research and service roles

Enjoy software and development!

Researching aspects of global SE and computer science education

THE SPREAD OF LATIN

Top 10 programming Language Trends for Developers to Follow in 2023

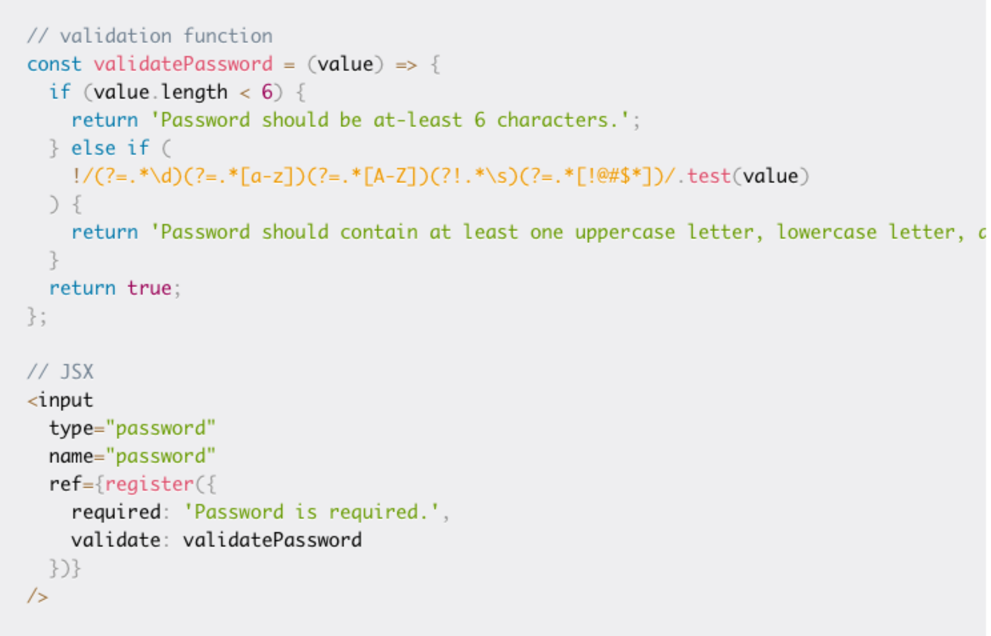


```
import React from "react";
import { useForm } from "react-hook-form";
import "./styles.css";

export default function App() {
  const { register, handleSubmit, errors } = useForm();

  const onSubmit = (data) => {
    console.log(data);
  };

  return (
    <div className="App">
      <form onSubmit={handleSubmit(onSubmit)}>
        <div className="form-control ">
          <label>Email</label>
          <input
            type="text"
            name="email"
            ref={register({
              required: true,
              pattern: /^[^@ ]+@[^@ ]+\.[^@ .]{2,}$/
            })}
          />
          {errors.email && errors.email.type === "required" && (
            <p className="errorMsg">Email is required.</p>
          )}
          {errors.email && errors.email.type === "pattern" && (
            <p className="errorMsg">Email is not valid.</p>
          )}
        </div>
        <div className="form-control">
          <label>Password</label>
          <input
            type="password"
            name="password"
            ref={register({ required: true, minLength: 6 })}
          />
          {errors.password && errors.password.type === "required" && (
            <p className="errorMsg">Password is required.</p>
          )}
          {errors.password && errors.password.type === "minLength" && (
            <p className="errorMsg">
              Password should be at-least 6 characters.
            </p>
          )}
        </div>
        <div className="form-control">
          <label></label>
          <button type="submit">Login</button>
        </div>
      </form>
    </div>
  );
}
```



```
// validation function
const validatePassword = (value) => {
  if (value.length < 6) {
    return 'Password should be at-least 6 characters.';
  } else if (
    !/(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?!.*\s)(?=.*[!@#$*])/.test(value)
  ) {
    return 'Password should contain at least one uppercase letter, lowercase letter, c
  }
  return true;
};

// JSX
<input
  type="password"
  name="password"
  ref={register({
    required: 'Password is required.',
    validate: validatePassword
  })}
/>
```

What does this code do and

tell me where the mistake is……

Taking Stock

The schedule for the course

Where are we now?

Class Representative

Perspectives Quiz

The Assessment Schedule

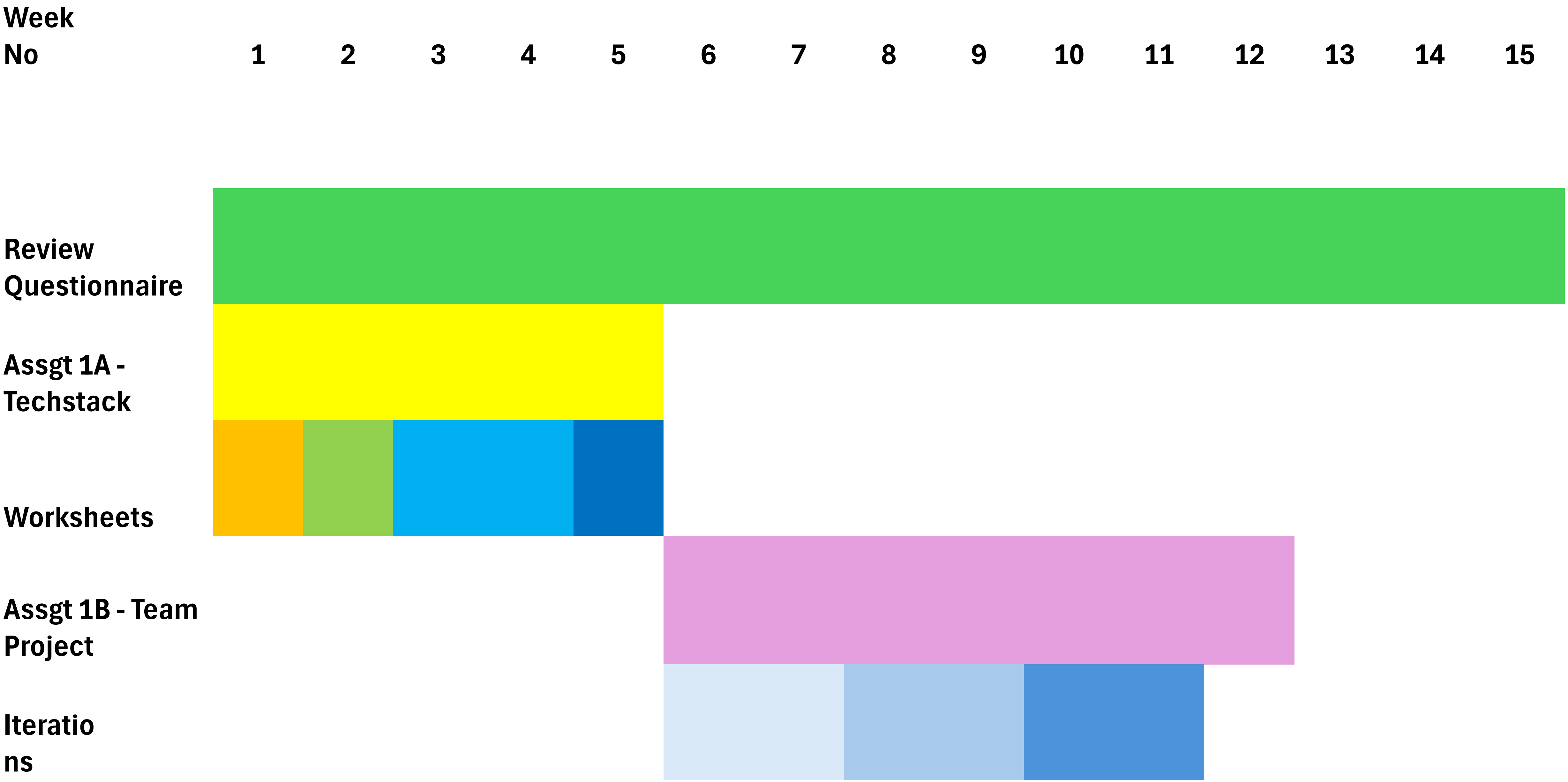
Progress – feedback, any issues?

Overview - what’s coming up?

The Lecture Schedule

How does it relate to the assessment?

Taking Stock



Why have this course – (Jim Buchan's) vision

The aim of the Software Development/Engineering majors is to give you opportunities to build the **knowledge**, **capabilities** and **attitudes** to become a **good software engineer**

So, what capabilities and attitudes does a good software engineer have?

Many! That's why good ones are in such demand..

The important thing is that you can apply these capabilities and attitudes to collaborate with a team of others to ….

…create, deploy and maintain high quality software that is of value to some users.

(Tony’s) views - Reflecting Over Time

acm INROADS

PAVING THE WAY TOWARD EXCELLENCE IN COMPUTING EDUCATION

https://inroads.acm.org/about.cfm

About *ACM Inroads*

The *ACM Inroads* magazine serves professionals interested in advancing computing education on a global scale.

Each issue of *ACM Inroads* presents the latest work, insights, and research in computing education as written *by* educators and professionals *for* educators. Authors represent an international community of scholars and professionals who reflect on and contribute to the computing profession.

The Association for Computing Machinery (ACM), the largest educational and scientific computing society in the world, publishes *ACM Inroads* quarterly.

My reflections here come from experiences expressed in roles as a regular Columnist and Associate Editor

Clear, T. (2003, Dec). **The Waterfall is Dead - Long Live the Waterfall!** *SIGCSE Bulletin, 35*(4), 13-14.

(Tony's) views – SE & *Tensions - 1*

Reflections on Software Development/Engineering Over time

- □ Requirements analysis
- □ Feasibility/Design
- □ Construction
- □ Implementation and testing

…core distinctions between **programming-in-the small** and **programming-in-the-large**. Key questions such as "what is programming?" come to mind. Is programming "the implementation of a design",

Clear, T. (2003, Dec). **The Waterfall is Dead - Long Live the Waterfall!** *SIGCSE Bulletin, 35*(4), 13-14.

(Tony’s) views – SE & *Tensions* - 2

Reflections on Software Development/Engineering Over time

So it seems to me that we have **a tension between four opposing forces**:

□ a **force for change** built upon an **initial and evolving visi**on, which drives the software process
□ a **commercial force for certainty** of cost and outcomes
□ a **project management force for certainty of delivery** against targets
□ a **professional force for delivering quality software**

Highsmiths's [1] **interaction, cooperation and collaboration** within the software process.

Clear, T. (2003, Dec). **The Waterfall is Dead - Long Live the Waterfall!!** *SIGCSE Bulletin, 35*(4), 13-14.

(Tony's) views – *Feature Driven Dev't -1*

"Information systems **success is achieved** when an information system is **perceived to be successful** by the stakeholders and other observers".

…Feature Driven Development [7] …delivering **scheduled releases** on a **regular set of cycles** for a demanding client operating in an entrepreneurial climate.

"…*improved the relationship* between our client and us by allowing the client to consider priorities and goals when planning each release, and accordingly **change their requirements specifications as the business evolves**" [2].

Clear, T. (2004). **Students Becoming Political and "Incorrect" Through Agile Methods**. *SIGCSE Bulletin, 36*(4), 13 - 15.

(Tony's) views – *Feature Driven Dev't - 2*

…**delivering functionality constantly** helped them "**better estimate** the cost of each feature, allow for project creep and to therefore have better control over the project as a whole".

… '**time-boxing**',(namely setting short time delimited windows within which results must be produced). ..“**came to realise that time boxes were about forcing tough decisions**”.

…customers and developers have **features as a common ground** in which to "discuss, debate and decide on critical product features" [4]. Certainly in the online talent agency project [2], **packaging four features into each release** enabled the client to **adjust priorities** within defined delivery windows.

Clear, T. (2004). **Students Becoming Political and "Incorrect" Through Agile Methods**. *SIGCSE Bulletin, 36*(4), 13 - 15.

For Total Projects

Figure 3.4 shows the distribution of work-effort among the standard technical tasks experienced by DP Services on previously completed total projects* where the same function was implemented as was designed.

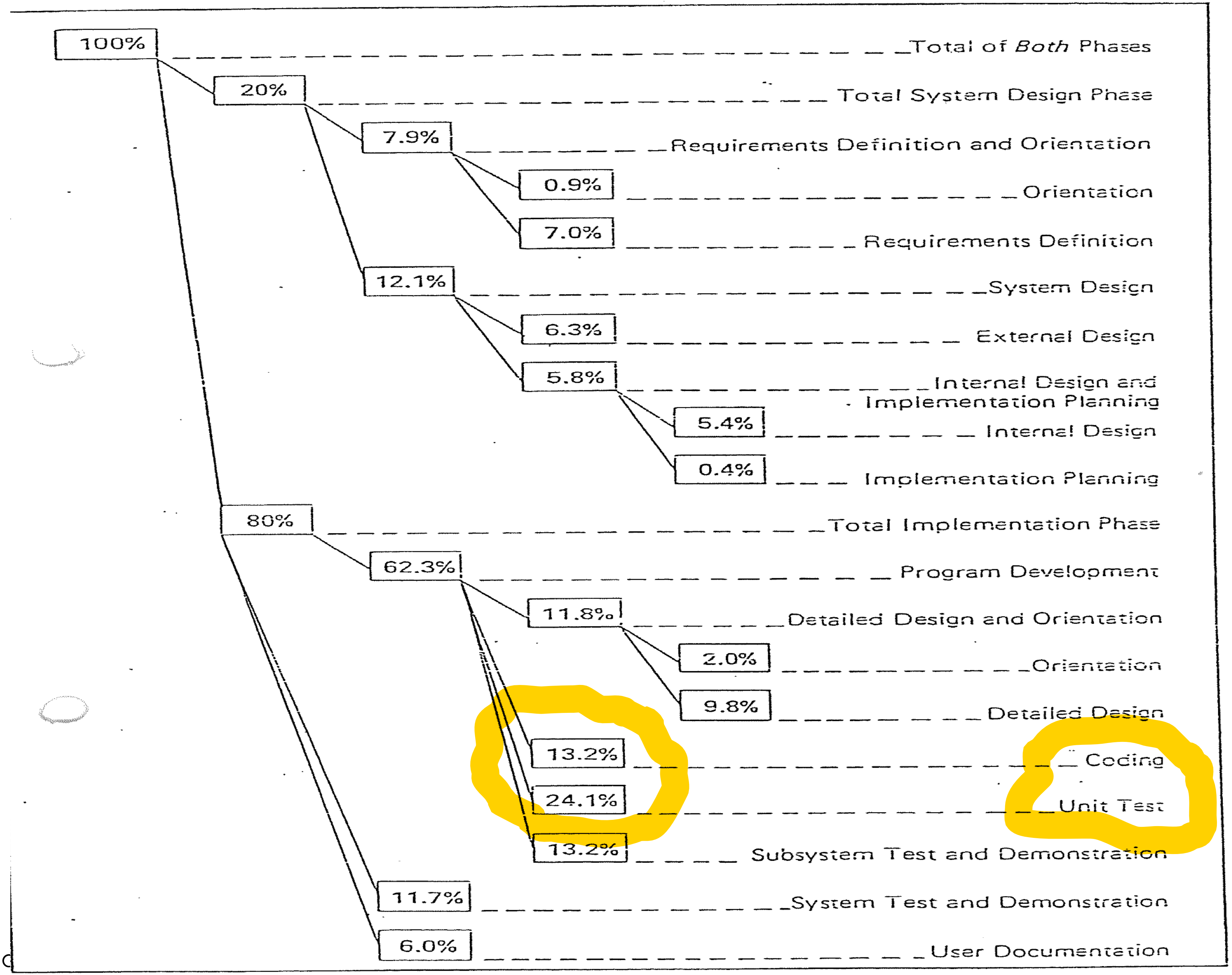

100% Total of *Both* Phases
20% Total System Design Phase
7.9% Requirements Definition and Orientation
0.9% Orientation
7.0% Requirements Definition
12.1% System Design
6.3% External Design
5.8% Internal Design and Implementation Planning
5.4% Internal Design
0.4% Implementation Planning
80% Total Implementation Phase
62.3% Program Development
11.8% Detailed Design and Orientation
2.0% Orientation
9.8% Detailed Design
13.2% Coding
24.1% Unit Test
13.2% Subsystem Test and Demonstration
11.7% System Test and Demonstration
6.0% User Documentation

Figure 3.4 Total Project Distribution

(Tony's) views - *Documentation- 1*

the views of Naur … the notion of **programming as "theory building"**, during which the programming team develops a jointly owned "theory of the world" to become frozen into software.

…**documentation as a secondary construct** to the programmers' internalised theory of the program or system, and based upon this "Theory Building View, **for the primary activity of the programming there can be no right method**"[2], since the creative process of theory building is inherently not a method or rule driven activity.

Clear, T. (2003). **Documentation and Agile Methods: Striking a Balance.** *SIGCSE Bulletin, 35*(2), 12 - 13.

(Tony’s) views - *Documentation*- 2

…documentation is not necessary **if the programming team can jointly own and hold the theory of the world in their head**.

…“the code, the code and nothing but the code” as the key artifact from a software project.

documentation as something external, something "other" than the primary work of coding, since it is only through the coding that the theory becomes encapsulated.

Ambler [5] suggests **two primary reasons for documentation**, namely that we should model (or document) **to communicate**, **or** model (or document) **to understand**.

Clear, T. (2003). **Documentation and Agile Methods: Striking a Balance.** *SIGCSE Bulletin, 35*(2), 12 - 13.

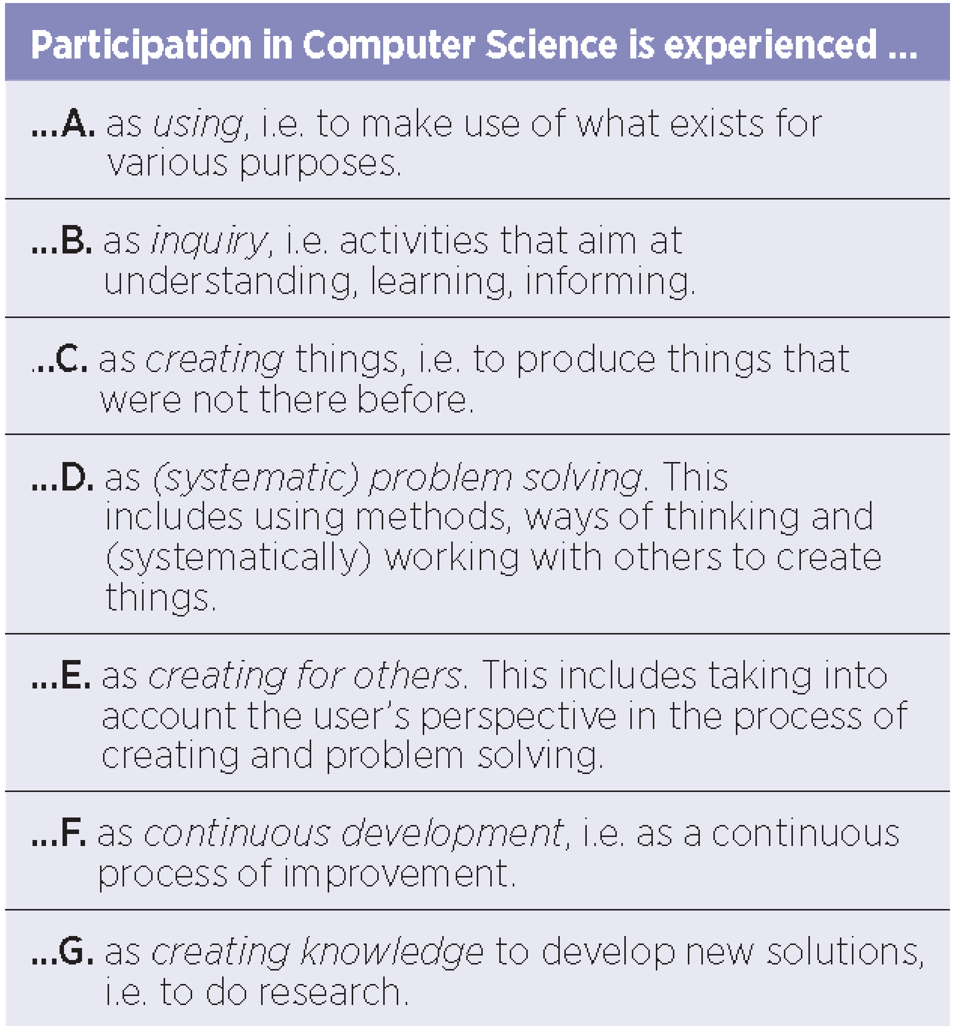
Working Globally in Software Engineering

Student Conceptions of the Discipline?

What does it mean to develop software as a professional software engineer?

Peters, A., Hussain, W., Cajander, A., Clear, T., & Daniels, M. (2015). **Preparing the Global Software Engineer**. In M. Nordio & B. Al-Ani (Eds.), *Proceedings 2015 IEEE 10th International Conference on Global Software Engineering* (pp. 61-70). IEEE. https://doi.org/10.1109/ICGSE.2015.20

Clear, T. (2016). THINKING ISSUES: **Computer science education---: challenging thinking in the small?** *ACM Inroads, 7*(2), 31-33. https://doi.org/10.1145/2927016



Participation in Computer Science is experienced ...
...A. as *using*, i.e. to make use of what exists for various purposes.
...B. as *inquiry*, i.e. activities that aim at understanding, learning, informing.
..C. as *creating* things, i.e. to produce things that were not there before.
...D. as *(systematic) problem solving*. This includes using methods, ways of thinking and (systematically) working with others to create things.
...E. as *creating for others*. This includes taking into account the user's perspective in the process of creating and problem solving.
...F. as *continuous development*, i.e. as a continuous process of improvement.
...G. as *creating knowledge* to develop new solutions, i.e. to do research.

Figure 1: Categories Describing Qualitatively Different Ways of Experiencing The Phenomenon Participation in CS/IT [10]

Working Globally in Software Engineering – Maturing?

Student Conceptions of the Discipline?

What does it mean to develop software?
as a professional software engineer?

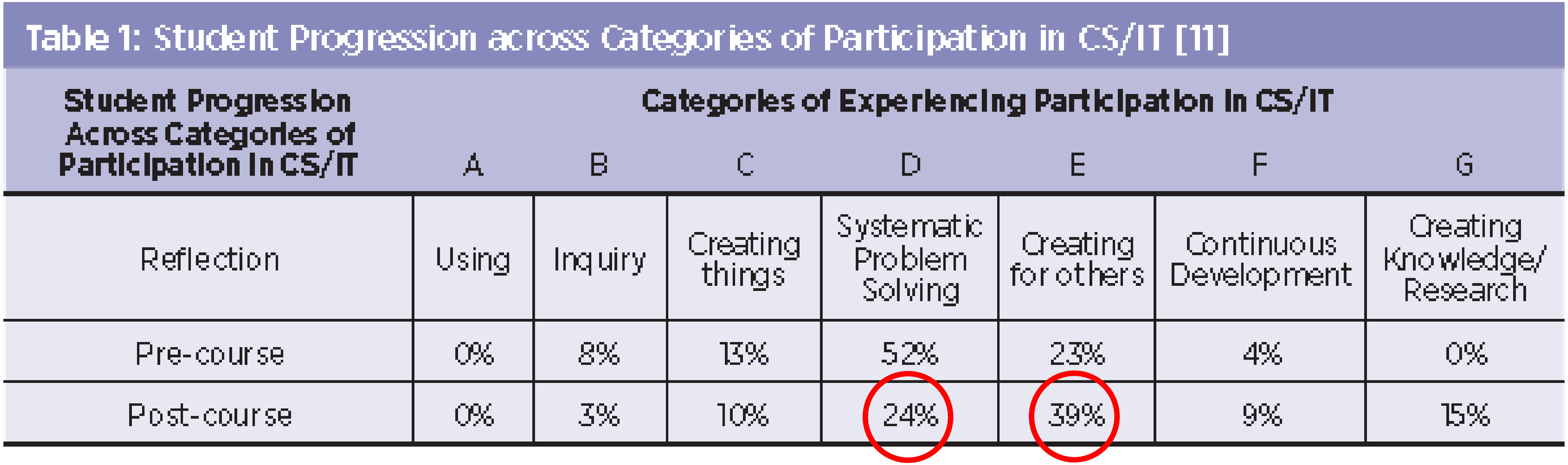


Table 1: Student Progression across Categories of Participation in CS/IT [11]

Student Progression Across Categories of Participation in CS/IT	Categories of Experiencing Participation in CS/IT						
	A	B	C	D	E	F	G
Reflection	Using	Inquiry	Creating things	Systematic Problem Solving	Creating for others	Continuous Development	Creating Knowledge/ Research
Pre-course	0%	8%	13%	52%	23%	4%	0%
Post-course	0%	3%	10%	24%	39%	9%	15%

Clear, T. (2016). THINKING ISSUES: **Computer science education---: challenging thinking in the small?** *ACM Inroads, 7*(2), 31-33. https://doi.org/10.1145/2927016

(Tony's) views – *Dispositions and Agility - 1*

"..*agilisme*"… core principle of agility from the agile manifesto [9] states that "***Our highest priority is to satisfy the customer through early and continuous delivery of valuable software***."

Implication **by a high performing team**

Unpacking the **three key aspects** then of agile development they can be said to comprise: 1) **customer satisfaction**, 2) **delivery of working software**, and 3) **provision of value**..

AUT Students and Healthy Together Technology

This year, the Healthy Together Technology team has been supporting a group of AUT Bachelor of Computer and Information Management students.

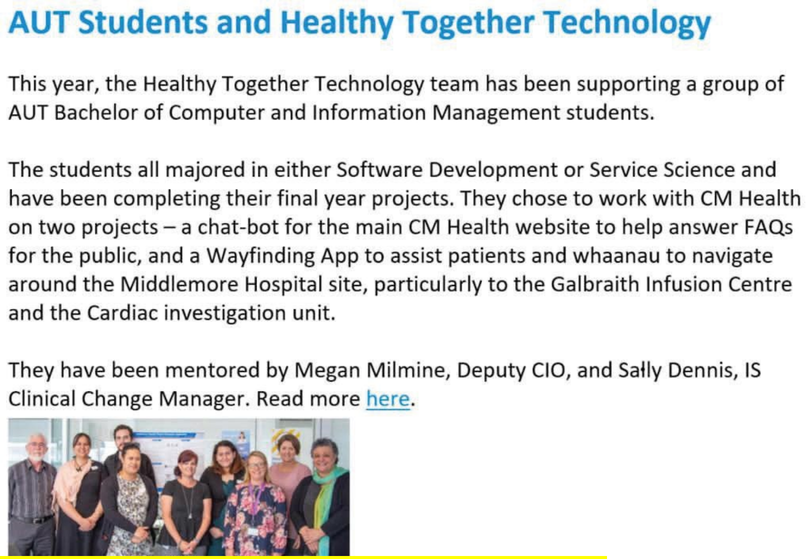
The students all majored in either Software Development or Service Science and have been completing their final year projects. They chose to work with CM Health on two projects – a chat-bot for the main CM Health website to help answer FAQs for the public, and a Wayfinding App to assist patients and whaanau to navigate around the Middlemore Hospital site, particularly to the Galbraith Infusion Centre and the Cardiac investigation unit.

They have been mentored by Megan Milmine, Deputy CIO, and Sally Dennis, IS Clinical Change Manager. Read more here.

Clear, T. (2021). THINKING ISSUES: **Is Agility a Disposition and Can it be Taught?** . *ACM Inroads, 12*(1), 13-14. doi:10.1145/3447870

(Tony's) views – *Dispositions and Agility - 2*

…disposition "***concerns not what abilities people have***, *but* ***how people are disposed to us***e *those abilities*." [10] "*So here we are talking about a mindset and attitudinal dimensions, which raises the question can a disposition be taught or is it some innate part of a person's character?*"



AUT Students and Healthy Together Technology

This year, the Healthy Together Technology team has been supporting a group of AUT Bachelor of Computer and Information Management students.

The students all majored in either Software Development or Service Science and have been completing their final year projects. They chose to work with CM Health on two projects – a chat-bot for the main CM Health website to help answer FAQs for the public, and a Wayfinding App to assist patients and whaanau to navigate around the Middlemore Hospital site, particularly to the Galbraith Infusion Centre and the Cardiac investigation unit.

They have been mentored by Megan Milmine, Deputy CIO, and Sally Dennis, IS Clinical Change Manager. Read more here.

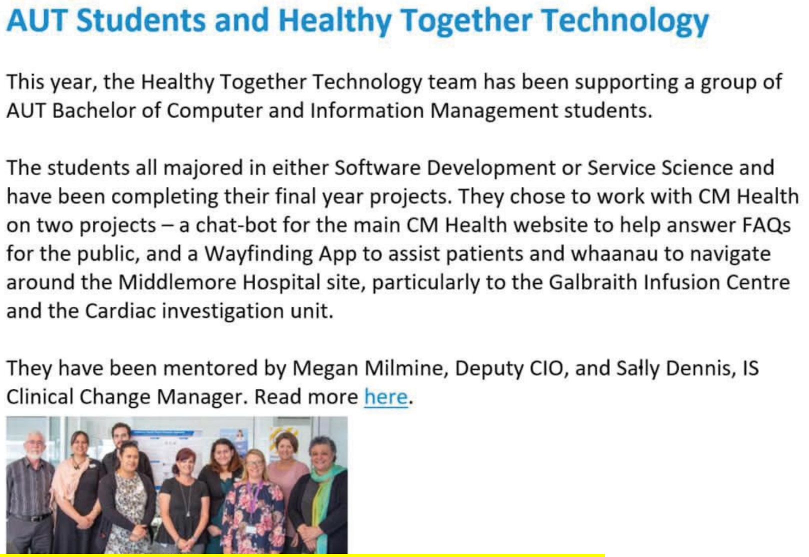
Clear, T. (2021). THINKING ISSUES: **Is Agility a Disposition and Can it be Taught**? . *ACM Inroads, 12*(1), 13-14. doi:10.1145/3447870

(Tony’s) views – *Dispositions and Agility* - 3

…producing a full quality assurance plan or change management plan when the project approach had not yet been determined, **did not make sense**, or as we discussed at the mid-project review

‘**did not produce value for the client**,’

team who **internalized the need for discretion and judgement** when deciding **which tasks and deliverables** contributed to providing value, **so made sense to produce,**
and at what time.



AUT Students and Healthy Together Technology

This year, the Healthy Together Technology team has been supporting a group of AUT Bachelor of Computer and Information Management students.

The students all majored in either Software Development or Service Science and have been completing their final year projects. They chose to work with CM Health on two projects – a chat-bot for the main CM Health website to help answer FAQs for the public, and a Wayfinding App to assist patients and whaanau to navigate around the Middlemore Hospital site, particularly to the Galbraith Infusion Centre and the Cardiac investigation unit.

They have been mentored by Megan Milmine, Deputy CIO, and Sally Dennis, IS Clinical Change Manager. Read more here.

Clear, T. (2021). THINKING ISSUES: **Is Agility a Disposition and Can it be Taught?** . *ACM Inroads, 12*(1), 13-14. doi:10.1145/3447870

DevOps, Agility and Dispositions…

Complementing the required knowledge and skills, **job ads also showed a desire for recruits who had specific** (Attributes, **Dispositions**, Attitude & Philosophy) detailed in Appendix 1.

Schussler observes that "***dispositions are different from knowledge and skills***" and

"***concerns not what abilities people have, but how people are disposed to use those abilities***" *[41].*

*The dispositions and attributes sought were **typically human and team centric**, demanding flexibility and adaptability*

*a **wider mindset** including customer/business awareness, relationship management, communication, general business acumen and respect in collaborative relationships*

- Clear, T. (2021). THINKING ISSUES: Is Agility a Disposition and Can it be Taught? . *ACM Inroads, 12*(1), 13-14. https://doi.org/10.1145/3447870
- Clear, T. (2017). Meeting Employers Expectations of DevOps Roles: Can Dispositions Be Taught? . *ACM Inroads, 8*(2), 19-21. https://doi.org/10.1145/3078298
- Hussain, W., Clear, T., & MacDonell, S. (2017). Emerging Trends for Global DevOps: A New Zealand Perspective. In D. Cruzes & A. Sharma (Eds.), *Proceedings 2017 IEEE 12th International Conference on Global Software Engineering* (pp. 21-30). IEEE. https://doi.org/10.1109/ICGSE.2017.16

Leadership Atributes…

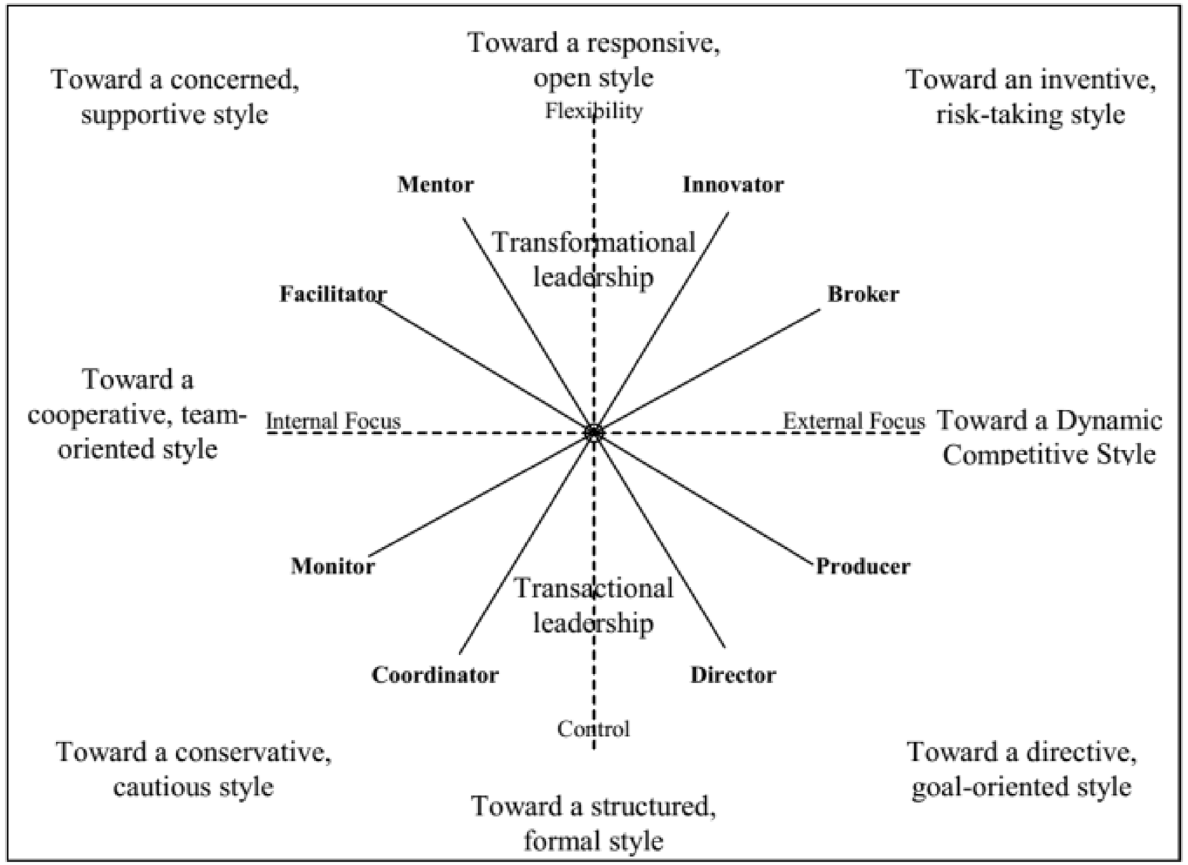

Toward a concerned, supportive style
Toward a responsive, open style
Flexibility
Toward an inventive, risk-taking style
Mentor
Innovator
Transformational leadership
Facilitator
Broker
Toward a cooperative, team-oriented style
Internal Focus
External Focus
Toward a Dynamic Competitive Style
Monitor
Producer
Transactional leadership
Coordinator
Director
Control
Toward a conservative, cautious style
Toward a structured, formal style
Toward a directive, goal-oriented style

Figure 5: Competing Values Framework of Leadership Roles [43]

…leadership attributes were actively sought. From our data we saw the strong emphasis on '**Transformational Leadership**' roles such as **Mentor, Facilitator, Innovator and Broker**. These stood in contrast to the more traditionally viewed expectations of technical employees engaged in projects and task related activities, better fitting a '**Transactional Leadership**' style.

Expectations of **taking responsibility for other team members** through **training and mentoring them** to develop new Knowledge Skills and Capabilities were very evident in the job ads.

- Hussain, W., Clear, T., & MacDonell, S. (2017). Emerging Trends for Global DevOps: A New Zealand Perspective. In D. Cruzes & A. Sharma (Eds.), *Proceedings 2017 IEEE 12th International Conference on Global Software Engineering* (pp. 21-30). IEEE. https://doi.org/10.1109/ICGSE.2017.16

Your SE journey in the BCIS ….

Programming 1	Coding basics. Good coding practice, code design. A few tools.
Programming 2	Coding in Object Oriented view. Good OO design and coding. IDE. Unit Tests
Program Design and Construction	Extending good coding practice – Design patterns, version control, architecture, …
Software Development Practice	Practice working in teams. Scrum -What and How. Own PO. Own Tech stack selection. Some tools
Contemporary Issues in Software Engineering	How to work in teams. Collaborate with External PO. The entire SDLC – from vision through evolving requirements to release. Specified tech stack. More tools. QA. More WoW. Hands on experience of how practices integrate with technologies and values to produce valuable high-quality results!
R&D Project	Working in teams for an EXTERNAL client. The entire SDLC. More tools, tech stack.

Problem-based learning

A real-world problem/opportunity for a client which can be addressed with a software product. You have to build the product as a team using good practice and tools. Planning-doing and reflecting on this drives the learning.

You understand theory and evidence - WHY you do something (WHAT) a particular way (HOW) and alternative options.

You practice to build capability and continuous learning from mistakes and reflection and team collaboration and technical skills and use of tools

You reflect on experience to compare theory to practice and evidence

You interact with a client/Product Owner

Developed and delivered in teams incrementally and iteratively using Agile informed ways of working and thinking and behaving

What work do we need to do and how and why?

Understand what to build – collaborating with a client - ongoing

Plan?

Build it

Deploy it

Maintain it

What work system, process, flow?

What are the values and principles that will guide how we decide to work, interact and behave?

What documentation is needed

How will this course help… a view of learning

- SE *language* used by other developers
- SE *concepts* used by other developers
- SE *Values* and *principles* that apply to many situations (patterns) and guide behaviour, interactions and work patterns
- SE *knowledge* – understand how practices and tools address SE problems, what is available and when to use them
- SE skills to *use* the tools, practices effectively

You will learn best when you are **motivated**…

- You understand the purpose of the learning
- You have some choice in learning (interested)
- It is safe to ask questions and give opinions
- It is safe to make mistakes and learn from them
- You develop mental models – Experiences - Interrupt or reinforce your models
- You are exposed to a diversity of options, experiences and people – no one way is right
- You get the chance to teach others
- You APPLY theory and knowledge to make software and interact as a team
- The work is at a difficulty level where it is a struggle but you get the buzz of succeeding

My expectations of you

- Hard working – minimum 8 hours per week outside of class
- Curious – willing (enjoy) to ask questions
- Committed to understanding my expectations
- Honest with me and each other
- Committed to learning (not just a grade?)
- Respectful of me and each other
- Open to new ideas and ways of working
- Support each other – have empathy
- Willing to think and probe and critique and disagree
- Willing to compromise in teams

Assignments Drive your Learning

Ass 1A preparing for Software Development (20%) (*Individual*)

- Set up the tools an individual needs to support coding, good code craft, version control and unit testing
- Set up the tools needed to collaborate with a team to achieve product goals together

Sharing code – integrate code, review code,
Setup the tools needed to work with the selected Tech Stack (front-end/backend)
Set up tools to assure quality of product
Set up tools to deploy the product to the cloud
Set up tools to monitor and alert issues post deploy
Learn how to use the tools
Learn how to use the Tech Stack
Understand the product goals -> Product Backlog
Sprint 1 Goals -> Sprint Backlog

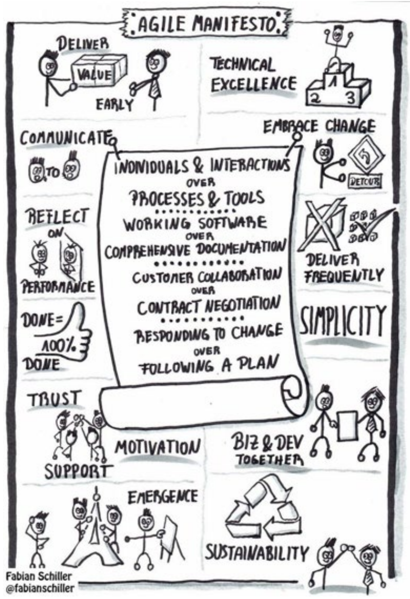
Submission in Tutorials weeks 1-5 (sign off by TA)
Evidence portfolio and demo

Ass1B Full SDLC full stack product Dev (50%) (*small team 4*)
Capability building by Developing a Product in a small team
Apply a new tech stack and tool set
Practice DevOps and Scrum WoW
Collaborate with a Product Owner and team
Three sprints to learn fast – fast feedback
Submit – reviews weeks 8,10,12 (tutorials)
Capability and learning Portfolio with Evidence
Product increments

Sprint 1 weeks 6 and 7
Sprint 2 weeks 8 and 9
Sprint 3 weeks 10 and 11

Ass 2 Knowledge Check (30%)
(*Individual, online questions*)
A set of questions about scenarios to confirm you have understood main language and principles
Sometime in Revision weeks (Faculty schedules)

Choose your team's WoW!



SEMAT (Software Engineering Methods and Theory)

Ways of Working

Agile ways of working

XP (eXtreme Programming)

Scrum

Kanban

Lean

DevOps

DevSecOps

Waterfall (plan-driven)

Our findings indicate that few participants run their projects in a purely agile or a purely traditional manner, and **most of them use home-grown hybrid development methods.**
Kuhrmann et al. (2022)

Tools to support WoW

Visual Studio. Git. GitHub.
JUnit. Cucumber. Selenium.
TravisCI. **Ansana**. Jira. AWS.
Heroku. Puppet. Ansible. **Maven**.
Docker. **Lint**. **SonarQube**. etc

User stories

Test driven Development

Continuous deployment

Code craft

Mob programming

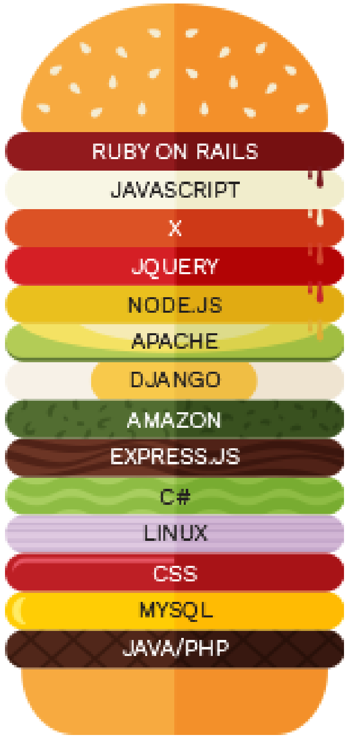
Continuous Integration

Pair programming

Code Review

Kuhrmann, M., Tell, P., Hebig, R., Klünder, J., Münch, J., Linssen, O., Pfahl, D., Feldere
Prause, C. R., MacDonell, S. G., Nakatumba-Nabende, J., Raffo, D., Beecham, S.,
Tüzün, E., López, G., Paez, N., Fontdevila, D., Licorish, S. A., Küpper, S., Ruhe, G., Kn

Learn how to learn a new Technology Stack… and supporting tools

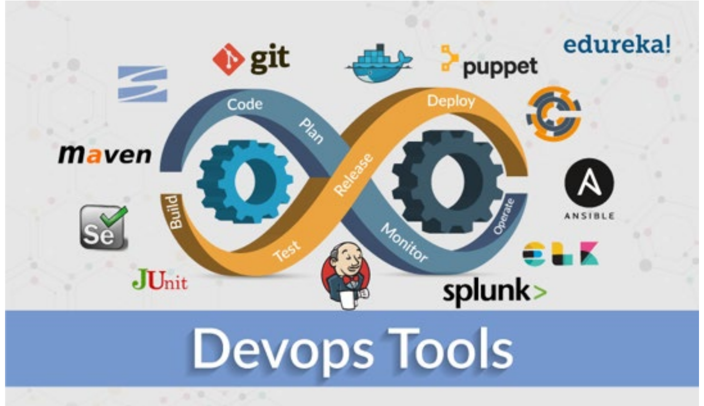


Javascript/Typescript based applications deployed in the cloud

- Backend – Nest
- Front end – NEXT.js
- Database – MongoDB?
- Other tools to optimize coding and quality
  - Visual Studio code, Eslinter, Prettier etc

Automated testing/checking from the start – tool pipeline

- GitHub for **version control, code sharing**, **code integration**
- GitHub Actions for (**CI/CD**) auto build and test, and deploy
- Jest for **end-to-end testing**
- Vercel for **cloud deployment**



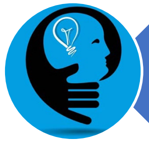
Diverse sources of knowledge/learning….

- **Online Tutorials** - Freecode.com, YouTube
- **Online documentation** - tools, libraries eg Docker, GitHub
- **Podcasts** Engineering Culture by InfoQ
- **Newsletters** - InfoQ
- **Meetup groups** Agile Auckland, DevOps, Ministry of testing,
- **Blogs** - Medium, Freecode.com,
- **GitHub Repositories** - Open source projects,
- **Company websites** -Google, Xero, Microsoft, Facebook, Netflix, Basecamp
- **Published research**– ACM, IEEExplore, SpringerLink, ScienceDirect
- **Online Q&A** -Stackoverflow
- **Guest speakers from industry**
- **Work in Industry or Open source projects**
- **Each other –** teaching and listening
- **Your lecturer**

Use of CANVAS and MS teams

- This semester ENSE701 will run as an **on-campus** course for **lectures** and **labs**
- Canvas will be used as a Repository of course content
    - For communicating announcements from time to time
    - And storing lecture recordings
- Teams channels will also be used for communicating with the product owner and the teaching team, so that they are all in the one place rather than lost in a deluge of emails
- As illnesses are still disrupting our lives… ☹
    - Teams may be used as a backup for meetings with students who have to isolate

Takeaways from today…a mental model of SE and this course



Engineering large software products is complex, can be seen from diverse perspectives and brings many challenges

(but it is great fun and rewarding and needed!)

There is no recipe – you need a box of tools and techniques to pull from and experiment with and adapt

This paper gives you the chance to extend your capability in ways of working, techniques and tools that focus on collaborative software development

Focus on building capability and knowledge as much as you can through doing and experimenting

You will need to keep on learning new techniques, ways of working together and tools from a variety of sources and sometimes just-in-time

Let's build a safe, collaborative, creative and curious learning environment

Questions and Comments….