CS304 SOFTWARE ENGINEERING
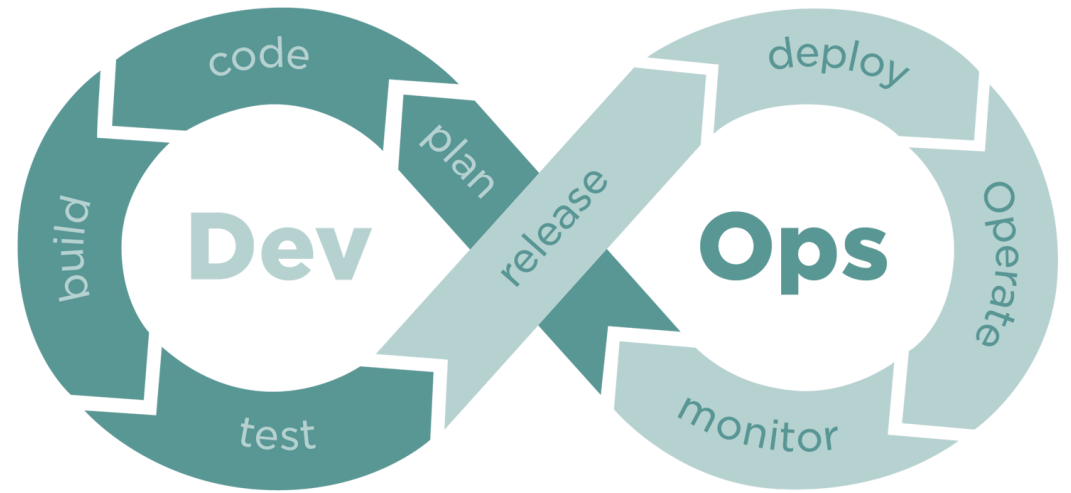
Yida Tao

taoyd@sustech.edu.cn

WHERE ARE WE NOW?
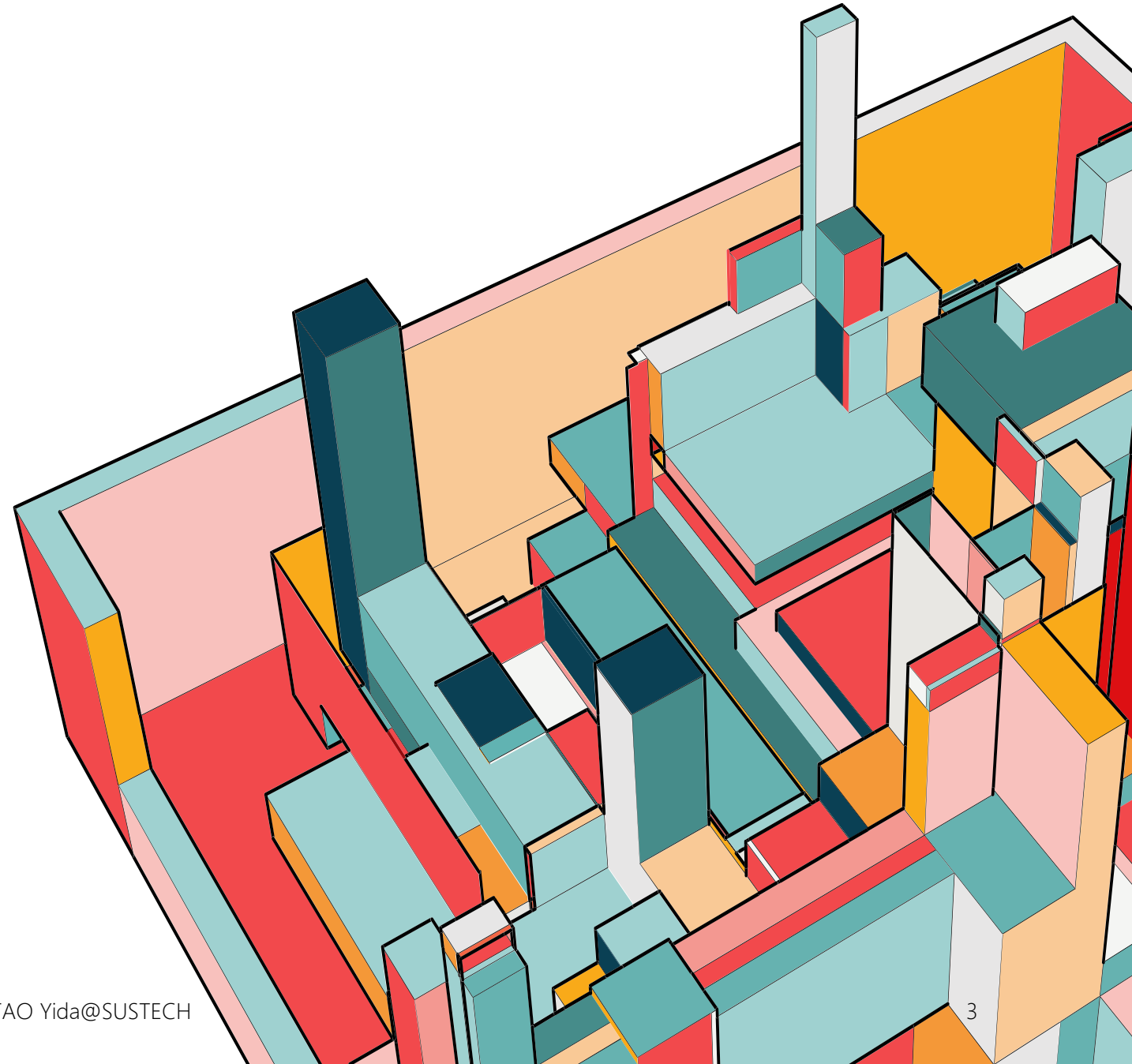
Plan
- **Requirement analysis**
- Design

LECTURE 4

- **Overview of software requirements**

- **Stakeholders**

- **Types of requirements**

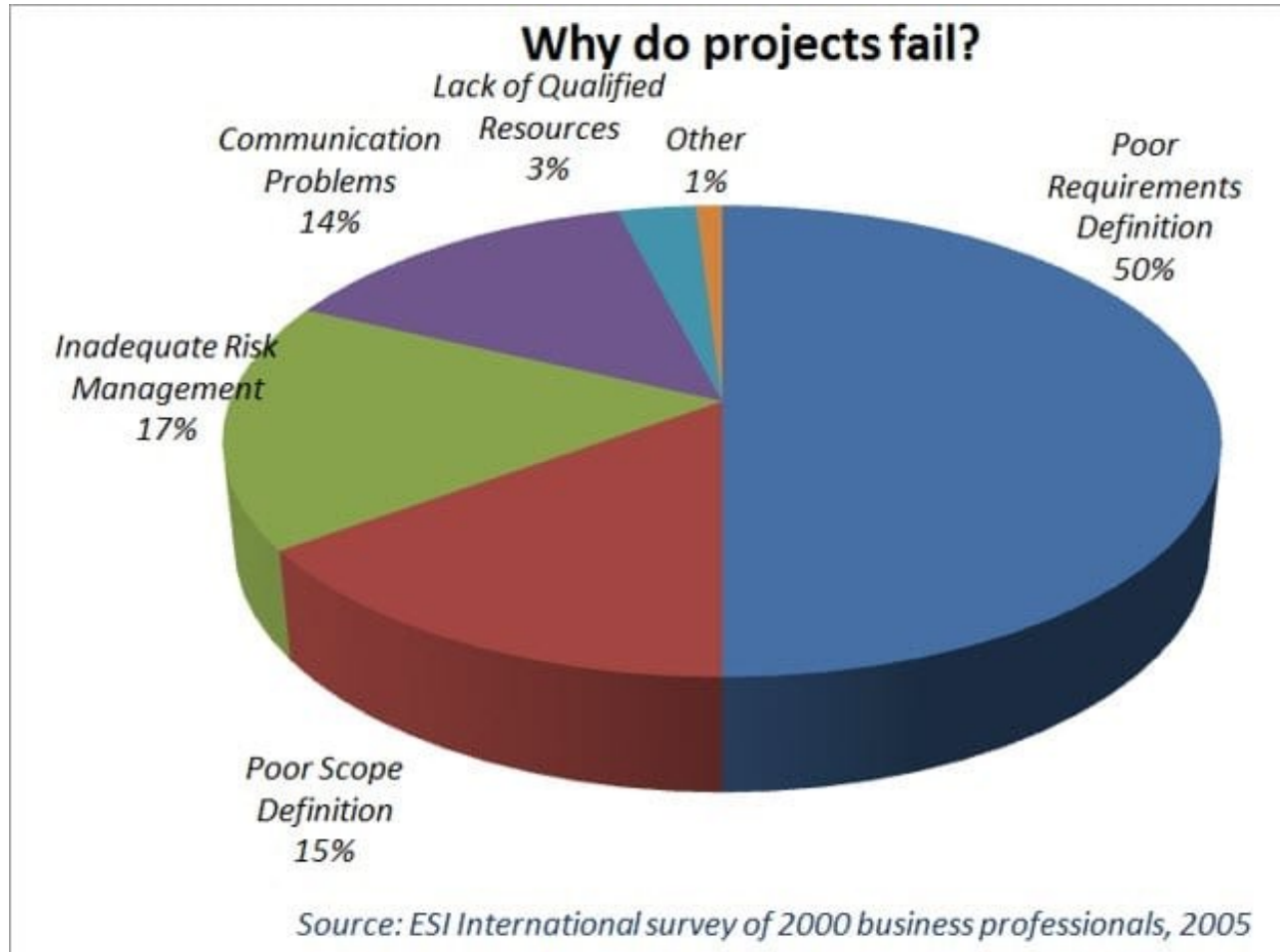- **Requirements analysis in Scrum**

WHAT IS SOFTWARE REQUIREMENTS?

- Requirements describe what the system should do, the services that it provides and the constraints on its operation, from the viewpoint of clients

- Software requirements establish the system's functionality, constraints, and goals.

Requirements: What the customers really needed

WHY ARE REQUIREMENTS IMPORTANT?

Why do projects fail?

- Lack of Qualified Resources 3%
- Communication Problems 14%
- Other 1%
- Poor Requirements Definition 50%
- Inadequate Risk Management 17%
- Poor Scope Definition 15%

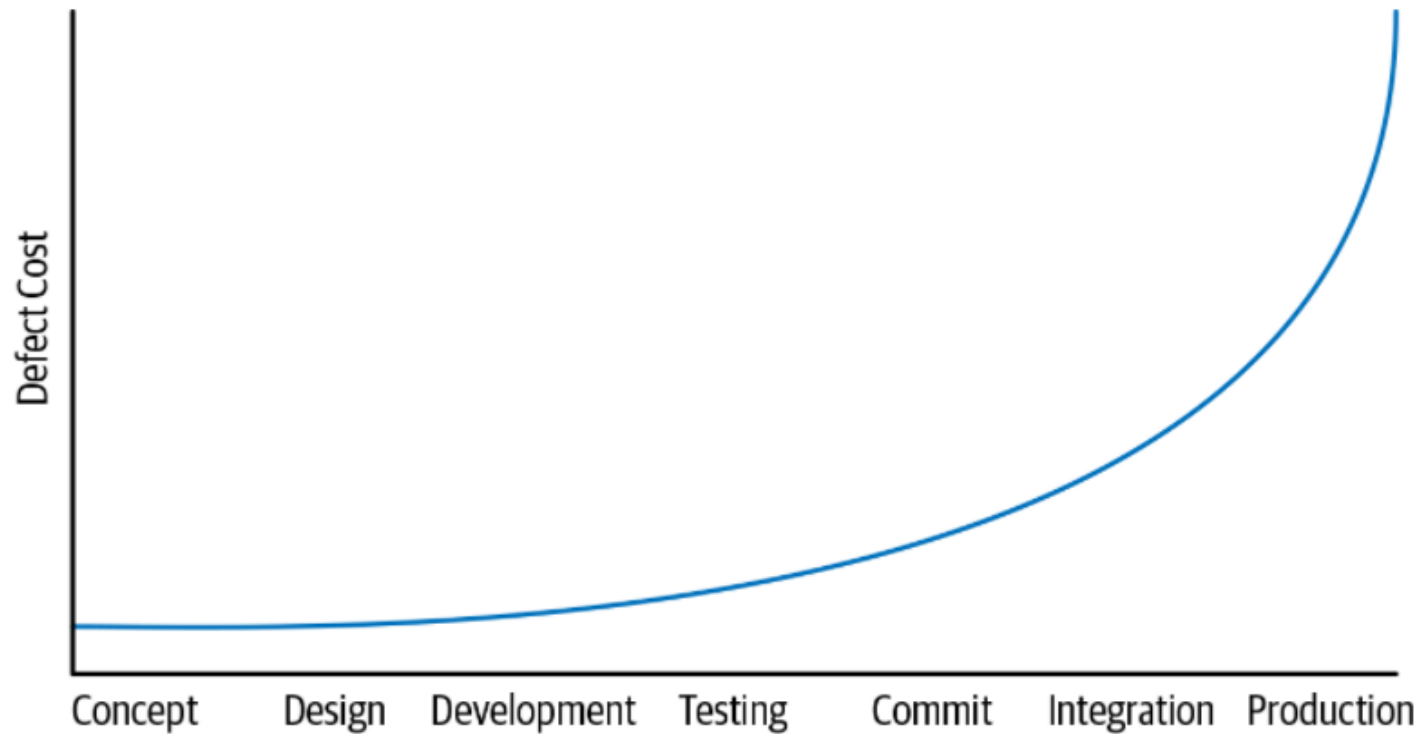Source: ESI International survey of 2000 business professionals, 2005

WHY ARE REQUIREMENTS IMPORTANT?

50% of source code defects can be attributed to deficiencies in requirements

Source: 《现代软件工程基础》

WHY ARE REQUIREMENTS IMPORTANT?

Figure 1-2. *Timeline of the developer workflow*

The later you identify requirement deficiencies, the higher the cost of resolving them

WHERE DO REQUIREMENTS COME FROM?

- Application domain
  - e.g., laws and regulations for banking software

- Stakeholders

PRIMARY STAKEHOLDERS

Primary stakeholders have a direct impact on your software project. They are people, groups, or organizations that have the strongest voice and can gain or lose their income.

- **Customers**: determine the main requirements and project scope and sign contracts with the main project performers.

- **Project Managers**: control the entire project creation process, considering the interests and needs of all stakeholders, create a solid product on time and within budget, supervise the project implementation processes, making necessary adjustments.

- **Business Analysts**: A BA team analyzes the customer's ideas, communicates with the development team, and determines project scope and requirements.

https://www.exposit.com/blog/primary-secondary-stakeholders-software-project/

PRIMARY STAKEHOLDERS

Primary stakeholders have a direct impact on your software project. They are people, groups, or organizations that have the strongest voice and can gain or lose their income.
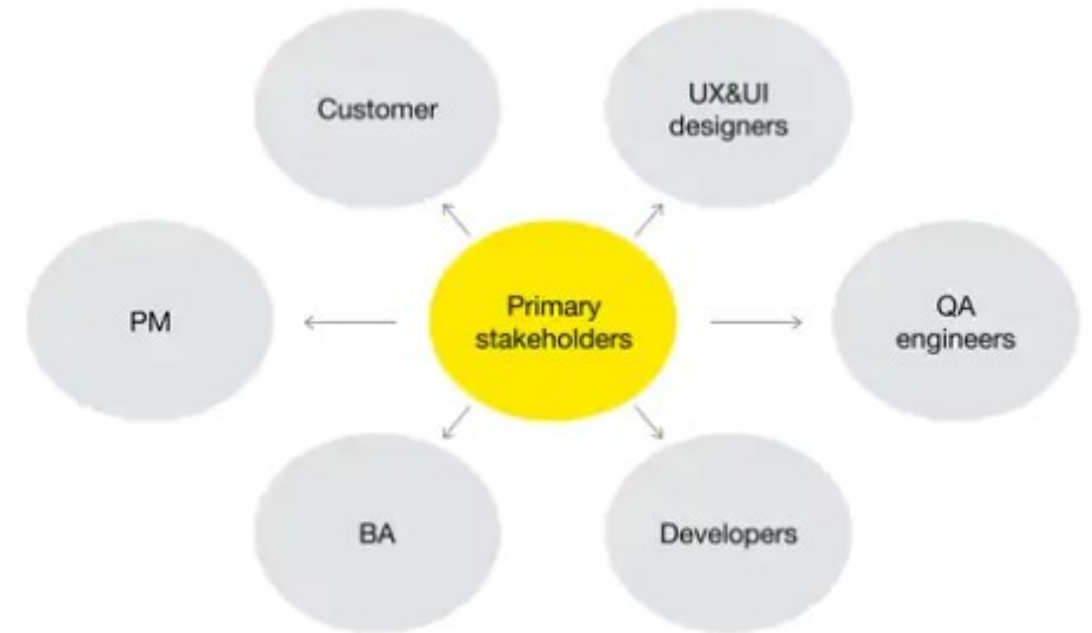
- **Development Team**: responsible for timely software delivery and estimation

- **Quality assurance (QA):** creating and performing tests, identifying errors, and providing feedback to verify that a final product meets a company's quality requirements.

- **UI&UX designers**: make the product interface user-friendly and understandable. Designers make customers get what they want quickly and easily.

https://www.exposit.com/blog/primary-secondary-stakeholders-software-project/
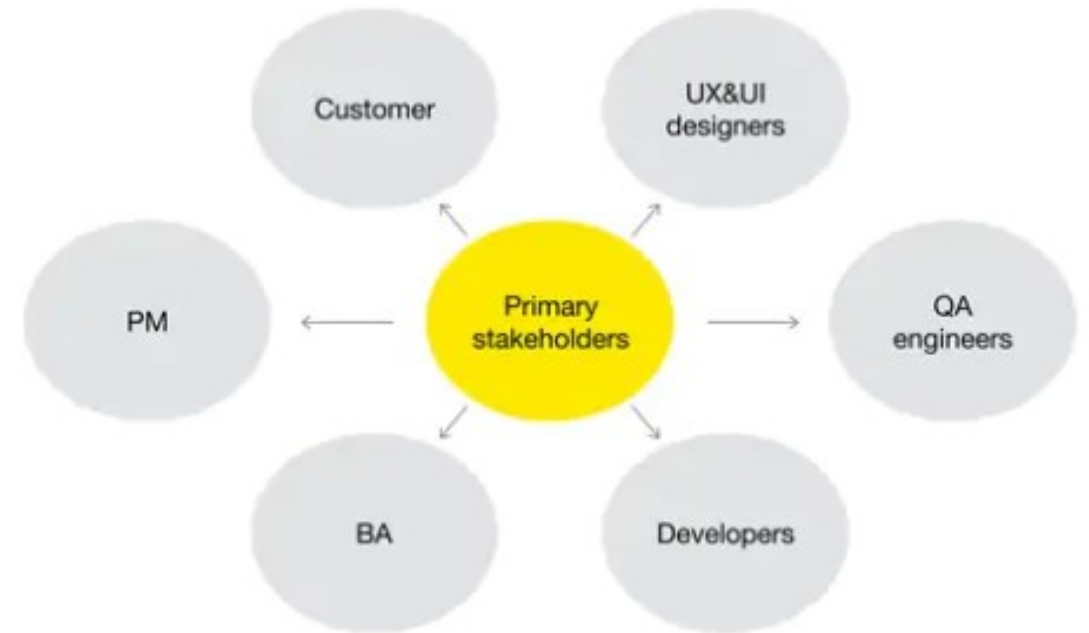
SECONDARY STAKEHOLDERS

Secondary stakeholders have an **indirect relationship** with a software development process. They do not have any direct engagement with a project or a company but can indirectly affect decisions related to your software product and its reputation.

- **End-users**: Users' needs and desires affect the design and functionality of the system. Users can be engaged as stakeholders in testing product beta-version to provide the initial feedback. They can point out missing features and contribute to user experience improvement.

- **Government**: Regulators adopt international standards that influence the development of a software product and impose fines for non-compliance with the rules.

- **Competitors**: Competitors always implement new features and create industry trends affecting the market. Thus, your competitors can influence the prioritization of tasks creating new unexpected challenges to respond to.
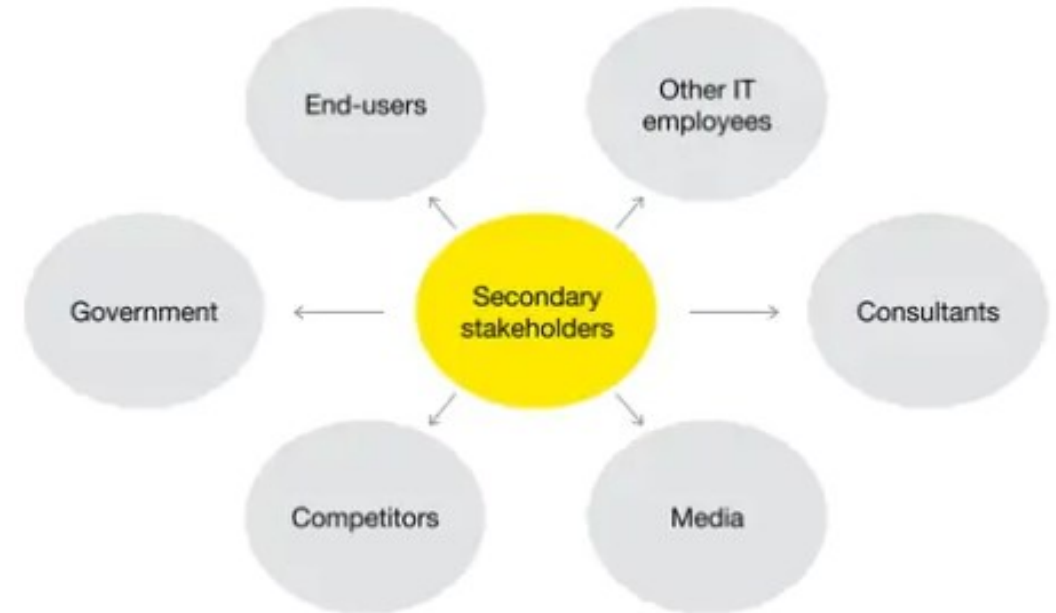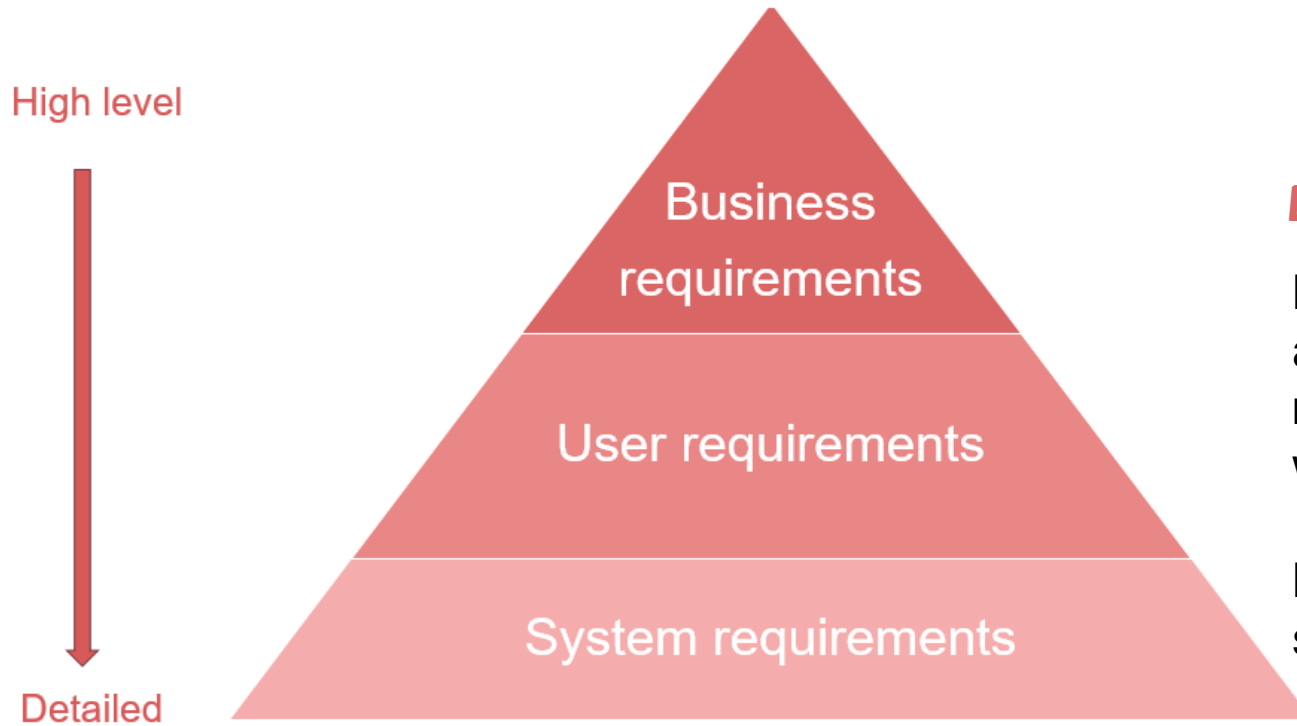

https://www.exposit.com/blog/primary-secondary-stakeholders-software-project/

LECTURE 4

- Overview of software requirements

- Stakeholders

- **Types of requirements**

- Requirements analysis in Scrum

CLASSIFICATION OF SOFTWARE REQUIREMENTS

High level

Detailed

Business requirements
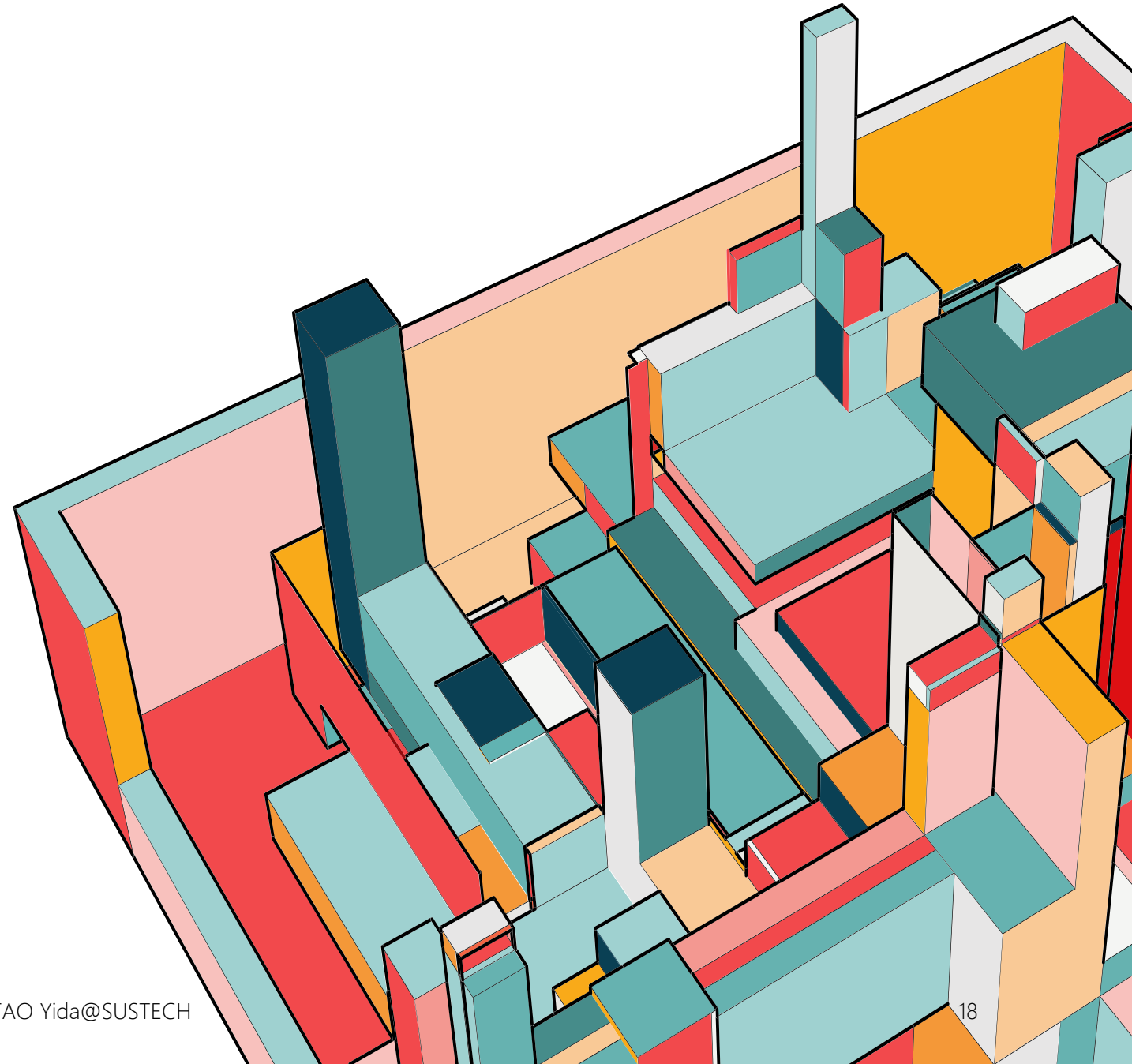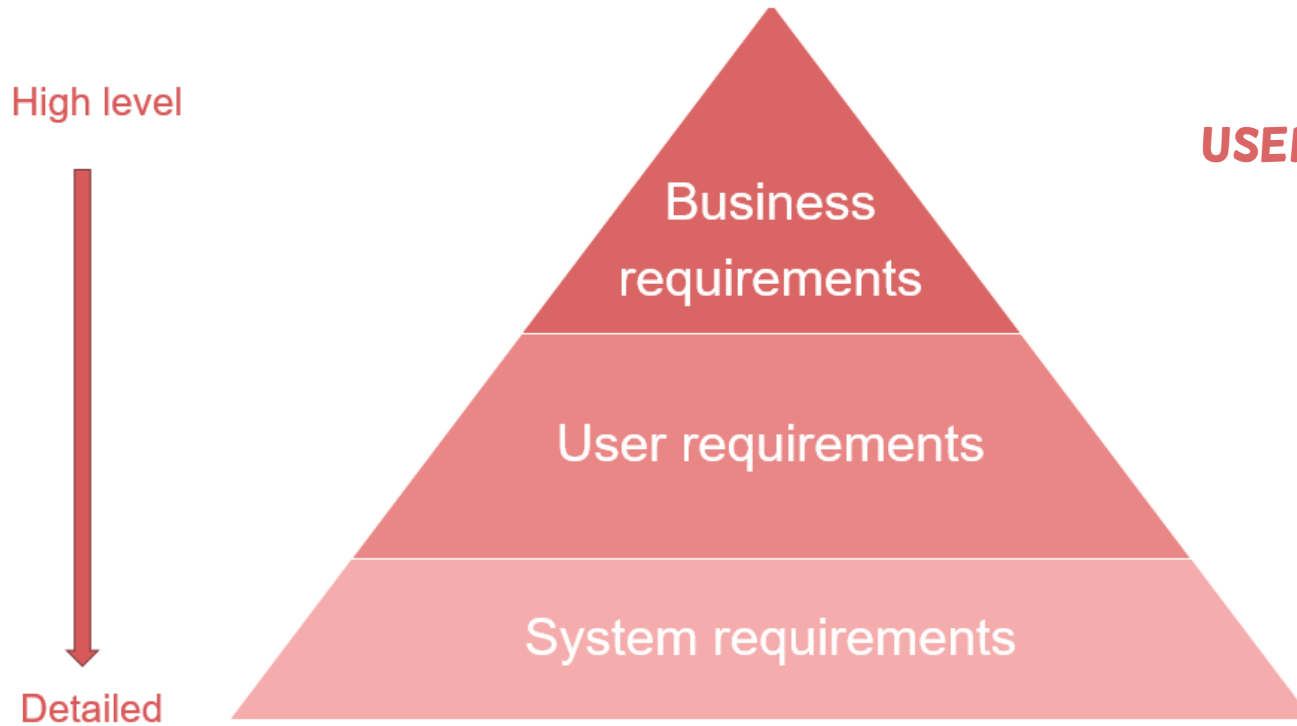
User requirements

System requirements

BUSINESS VIEW: WHY IS THE PROJECT NEEDED?

Business Requirements outline a general overview of a product, such as its primary usage, why it is needed, its scope and vision, what business benefits will be gain, intended audience or users, etc.

Business Requirements define the **WHY** behind a software project.

CLASSIFICATION OF SOFTWARE REQUIREMENTS

High level

Detailed

Business requirements
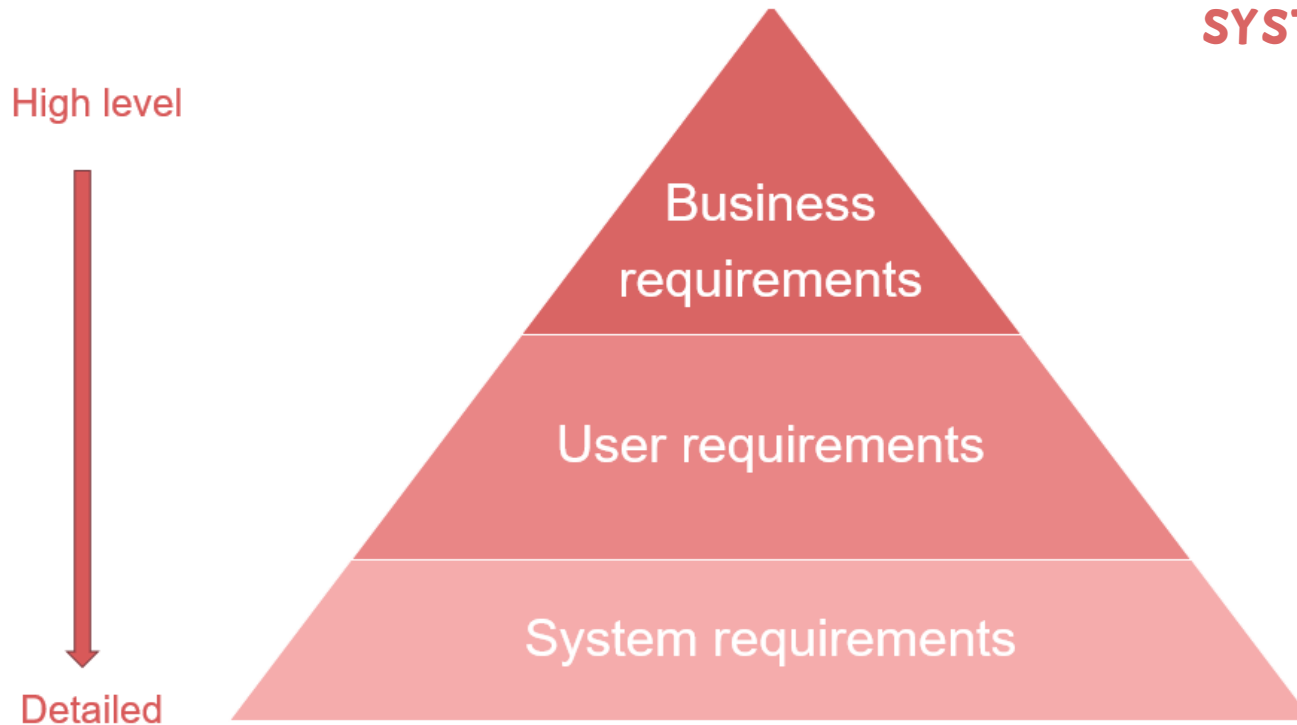
User requirements

System requirements

USER VIEW: WHAT DO USERS NEED THE SYSTEM TO DO?

User requirements are gathered using use case, user scenarios, and user stories

Often written in natural languages with NO technical details

User requirements describe the **WHO** of a software project.

CLASSIFICATION OF SOFTWARE REQUIREMENTS

High level

Detailed

Business requirements

User requirements

System requirements

SYSTEM VIEW: WHAT DOES THE SYSTEM NEED TO DO?

System requirements describe software as **functional** modules and **non-functional** attributes.
- Functional requirements
- Non-functional requirements

System requirements are written for developers with more formal format

System requirements dive into the **HOW** of a software project.

EXAMPLE

Business requirements, as visions (愿景), need to be iteratively decomposed (分解) and refined (精化), until achieving detailed and actionable system requirements

DECOMPOSED AND REFINED

VISION
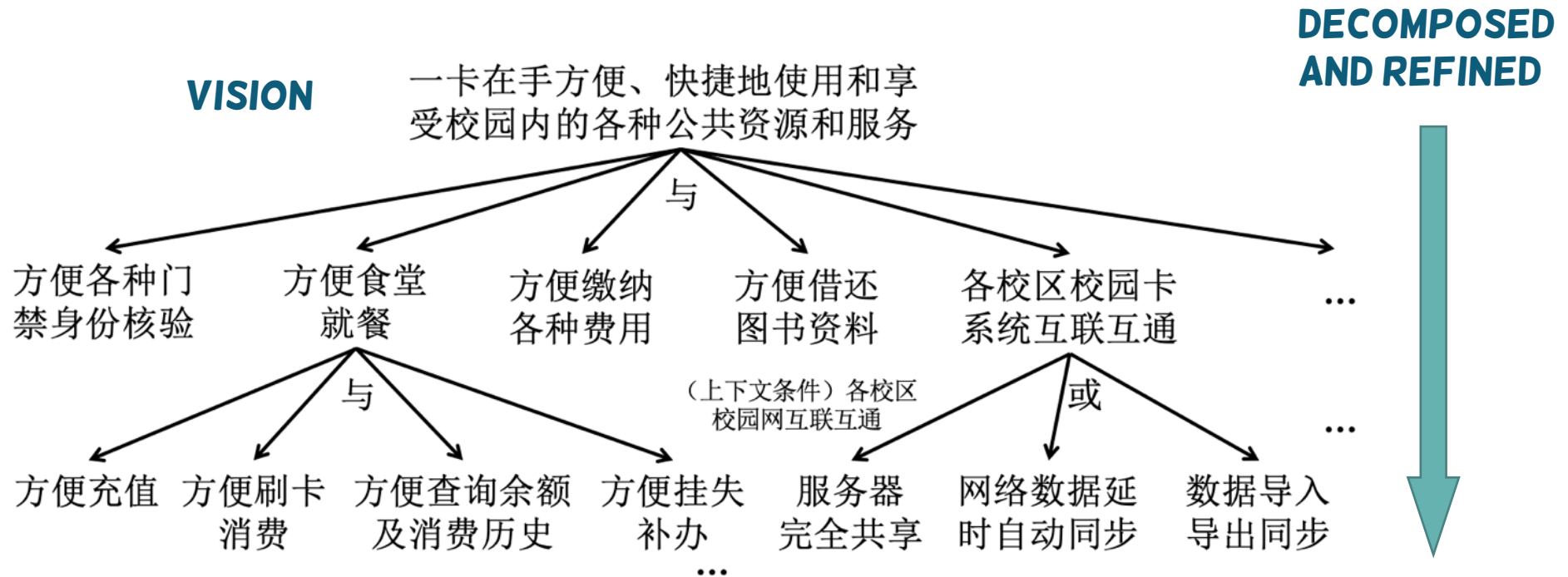
FUNCTIONAL REQUIREMENTS

- Functional requirements are basic functionalities that the system should offer.

- They are represented or stated in the form of **input** to be given to the system, the **operation** performed and the **output** expected.

- Functional requirements are basically the requirements stated by the user which one can see directly in the final product
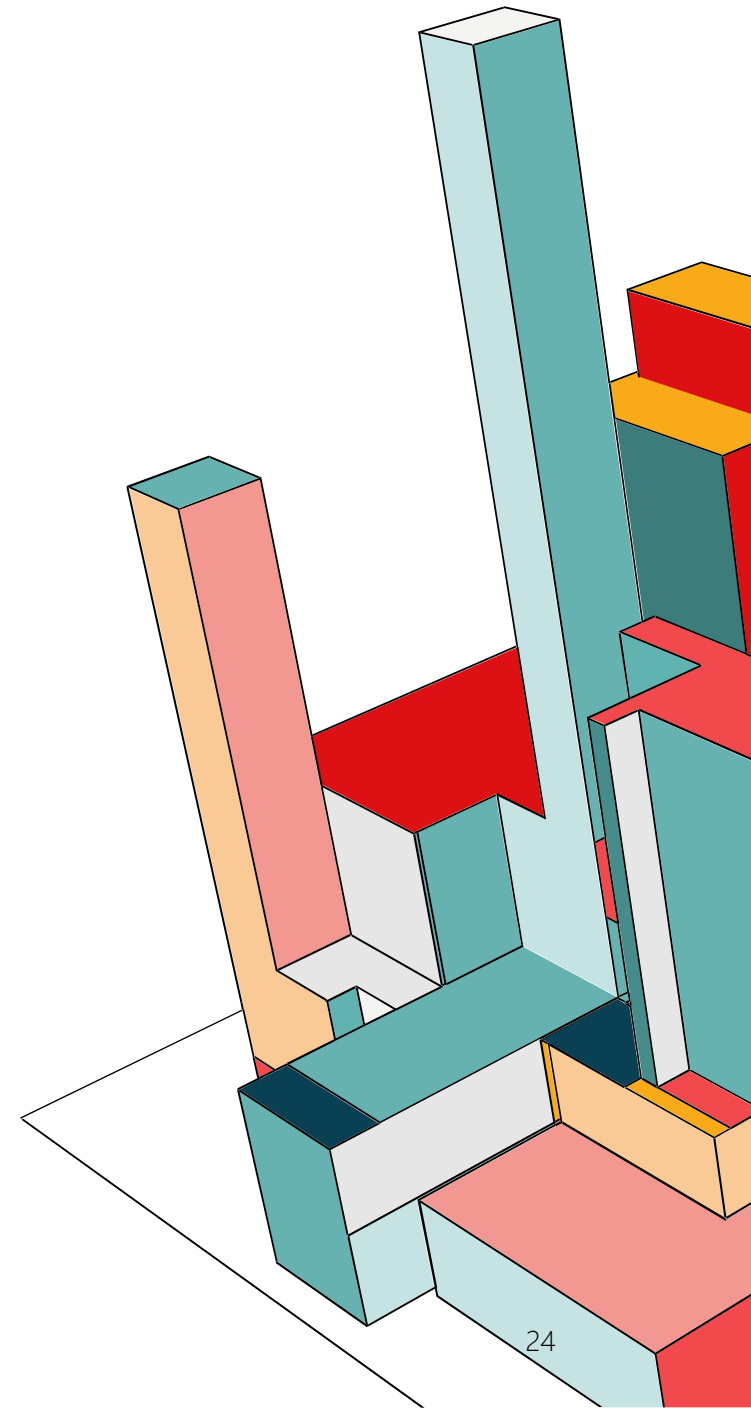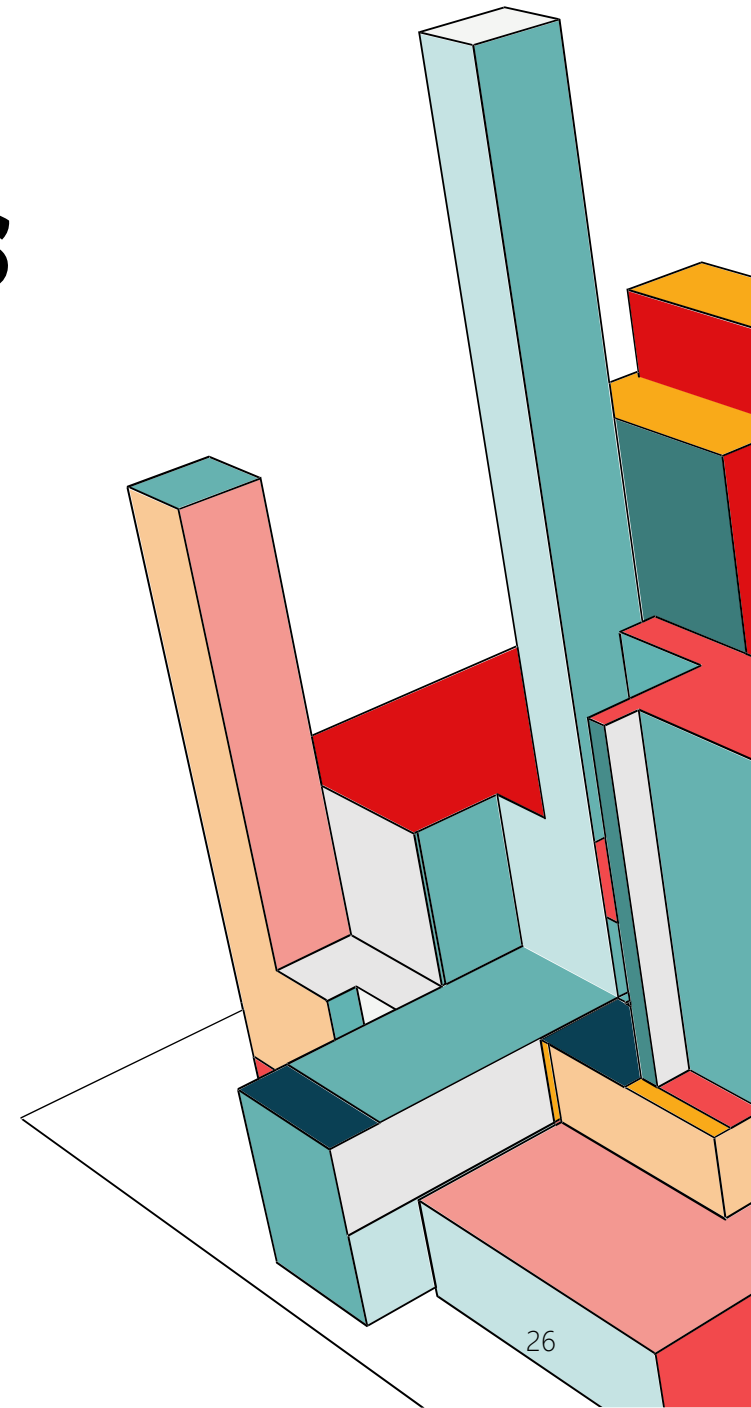
FUNCTIONAL REQUIREMENTS

Subtypes	Examples
系统应提供的服务 (service provided)	能够按照关键字检索图书 (searching books by keywords)
系统针对特定输入的响应 (actions for specific input)	对于格式不正确的身份证号进行提示并请用户重新输入 (re-enter ID for invalid input)
系统在特定情形下的行为 (behavior for specific context)	用户如果5分钟内没有操作，那么主界面自动进入锁定状态 (lock the screen if no user operations for 5 min)
系统不应做什么 (forbidden activity)	不允许尝试密码输入三次以上 (cannot input password over 3 attempts)

NON-FUNCTIONAL REQUIREMENTS

- Non-functional requirements are not related to the software's functional aspect

- Non-functional requirements specify the software's **quality attribute**.

- Non-functional requirements define the general characteristics, behavior of the system, and features that **affect the experience of the user**.

TAO Yida@SUSTECH

NON-FUNCTIONAL REQUIREMENTS

Subtypes	Examples
性能（performance）	联机刷卡应当在5秒内返回结果 (Online card swiping should return results within 5 seconds.)
可靠性（reliability）	系统的整体可靠性要达到99.99%以上
安全性（security）	系统应确保手机支付充值账户和密码不会被泄漏和盗用 (payment accounts and passwords will not be leaked or misused.)
易用性（usability）	用户根据提示学会手机支付充值的时间不超过10分钟 (Users should learn mobile payment within 10 minutes)
产品约束 (product constraints)	软件系统要在已有的几台服务器上运行并使用Linux系统 (Software needs to run on existing Linux servers)
过程约束 (process constraints)	软件系统应当在5个月内交付并严格遵循给定的过程规范 (Software should be delivered in 5 month)

LECTURE 4

- Overview of software requirements

- Stakeholders

- Types of requirements

- **Requirements analysis in Scrum**

TAO Yida@SUSTECH
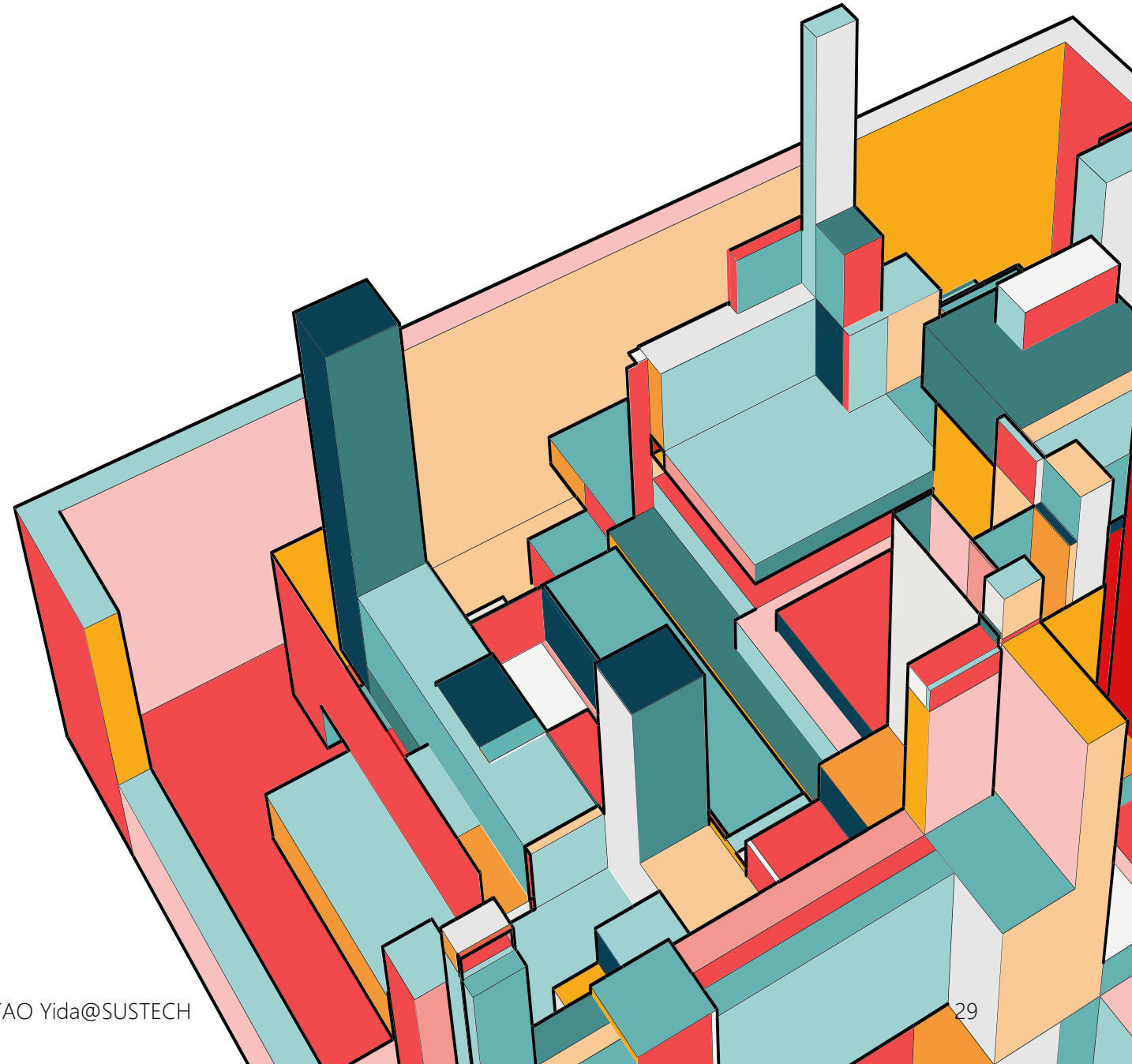
REQUIREMENT ANALYSIS IN SCRUM

- The product owner is responsible for talking to all of the stakeholders and gathering requirements.

- Based on the gathered information, the scrum team work together to come up with user stories, which are simple sentences that describe what the users need

COMMON FORMAT OF USER STORY

As a «role/profile»

I want to «action/activity»

so that «benefit/reason»

USER STORY EXAMPLES

https://www.youtube.com/watch?v=apOvF9NVguA

USER STORY EXAMPLES

As an
admin user
I want to
disable a user
So I can
prevent unauthorized logins by past
employees

As a
mobile app user
I want to
save all my data to the cloud
So I can
access it from another device

https://www.youtube.com/watch?v=apOvF9NVguA

MORE EXAMPLES

- As a participant, I want to register for an event so that I can attend it.

- As a sales professional, I want to generate reports so that I can take a decision on the marketing strategy for the upcoming quarter

- As a restaurant owner, I want to update my menu items in real-time on my website, so that my customers have accurate information.

- As an avid reader, I want to receive personalized book recommendations based on my reading history, so that I can discover new books that align with my interests.
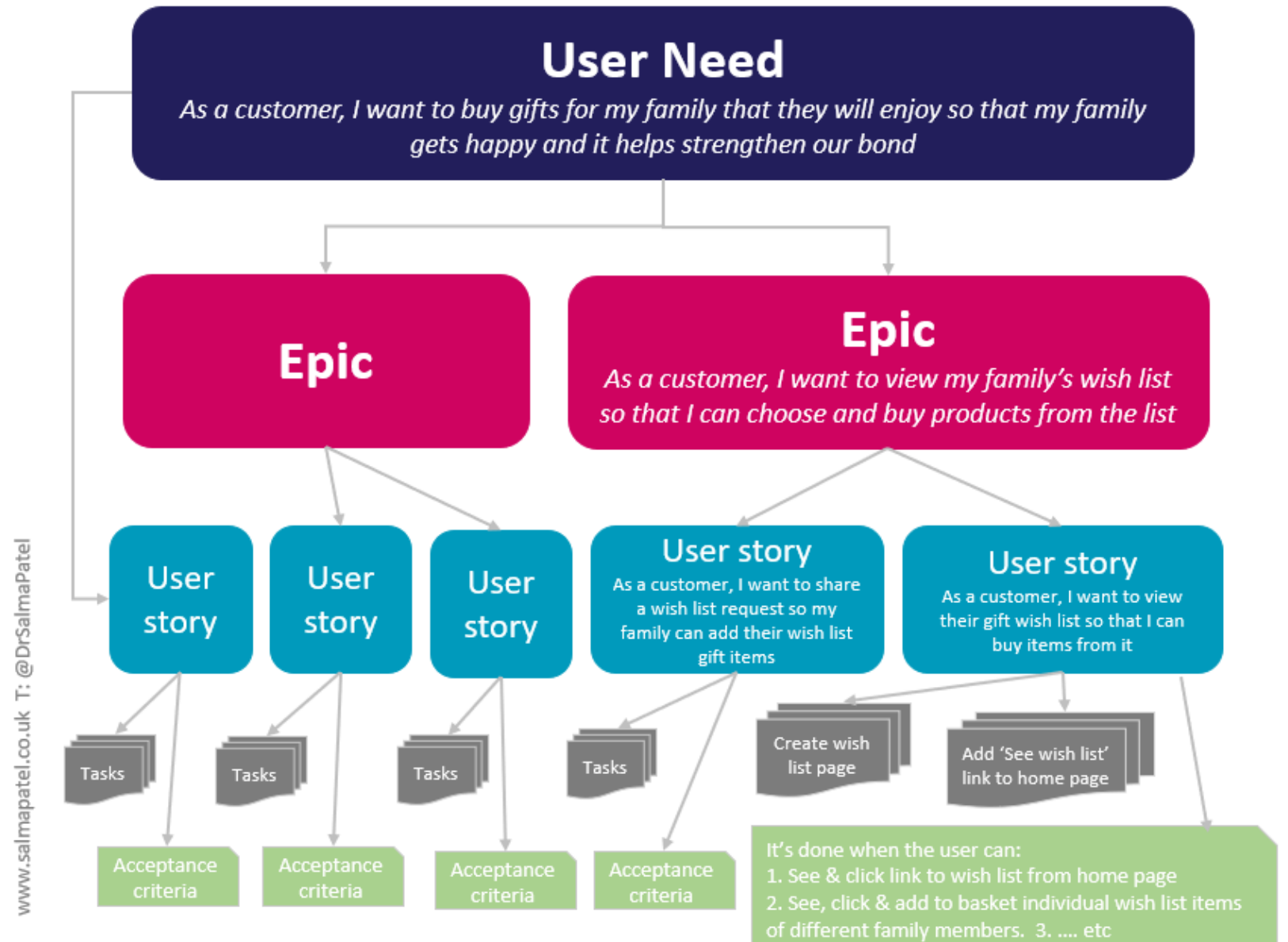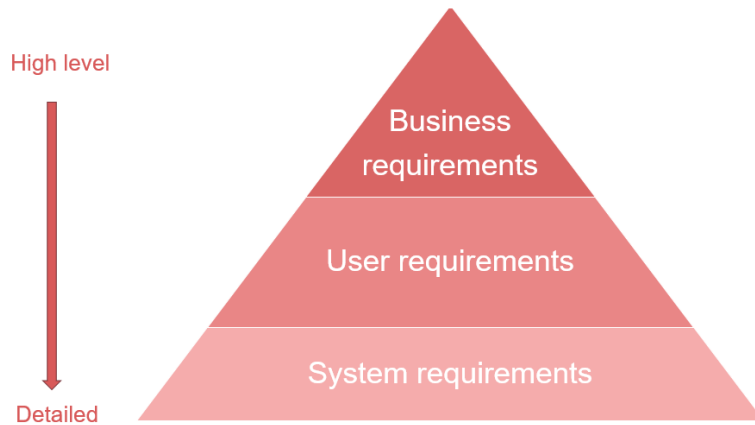
WHAT MAKES A GOOD USER STORY?

- Stories come from the perspective of **users**
- Stories create **business value** for the customers

Bad user story:
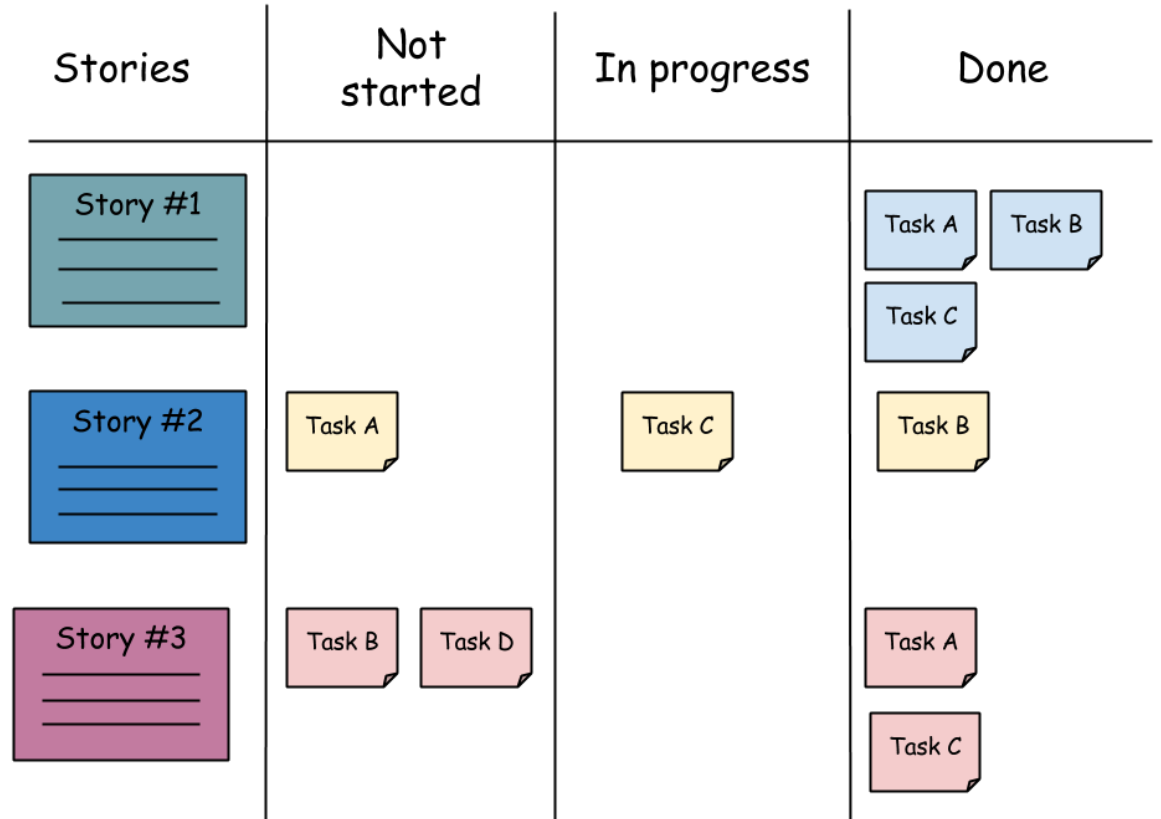"As a developer, I want a database with all the tables to model the data, so I can store information the application needs"

GRANULARITY OF USER STORY

FROM USER STORIES TO TASKS

- Tasks are used to break down user stories even further.

- Tasks are the smallest unit used in scrum to track work.

- A task should be completed by one person on the team

- A task typically takes ~1 day

FROM USER STORIES TO TASKS

As a customer, I want to be able to have wishlists so that I can come back to buy products later

User story

| Put 'Add to wishlist' button on each product page | Create new db to store wishlist items | Create page to display user's wishlist | Add 'View wishlist' link to homepage |

Tasks

Tasks should be small, manageable, and clear, so that everyone knows what need to be done.

FROM USER STORIES TO TASKS

USER STORIES FOR DIFFERENT SPRINT

Split along "Process Lines"

1 Start ───► End

| | Feature A | Feature B | Feature C | Feature D | Feature E | Feature F | Feature ... |

2 Split along "Level of Necessity"

Low

		Feature A	Feature B	Feature C	Feature D	Feature E	Feature F	Feature ...
Bare Necessity	Sprint 1	User Story, User Story, User Story	User Story	User Story, User Story	User Story, User Story, User Story	User Story	User Story	User Story, User Story
Usefulness	Sprint 2	User Story, User Story	User Story	User Story, User Story	User Story	User Story	User Story	User Story, User Story
Delightfulness	Sprint N	User Story	User Story	User Story	User Story	User Story	User Story	User Story

High

HOW TO KNOW IF A USER STORY IS COMPLETE?
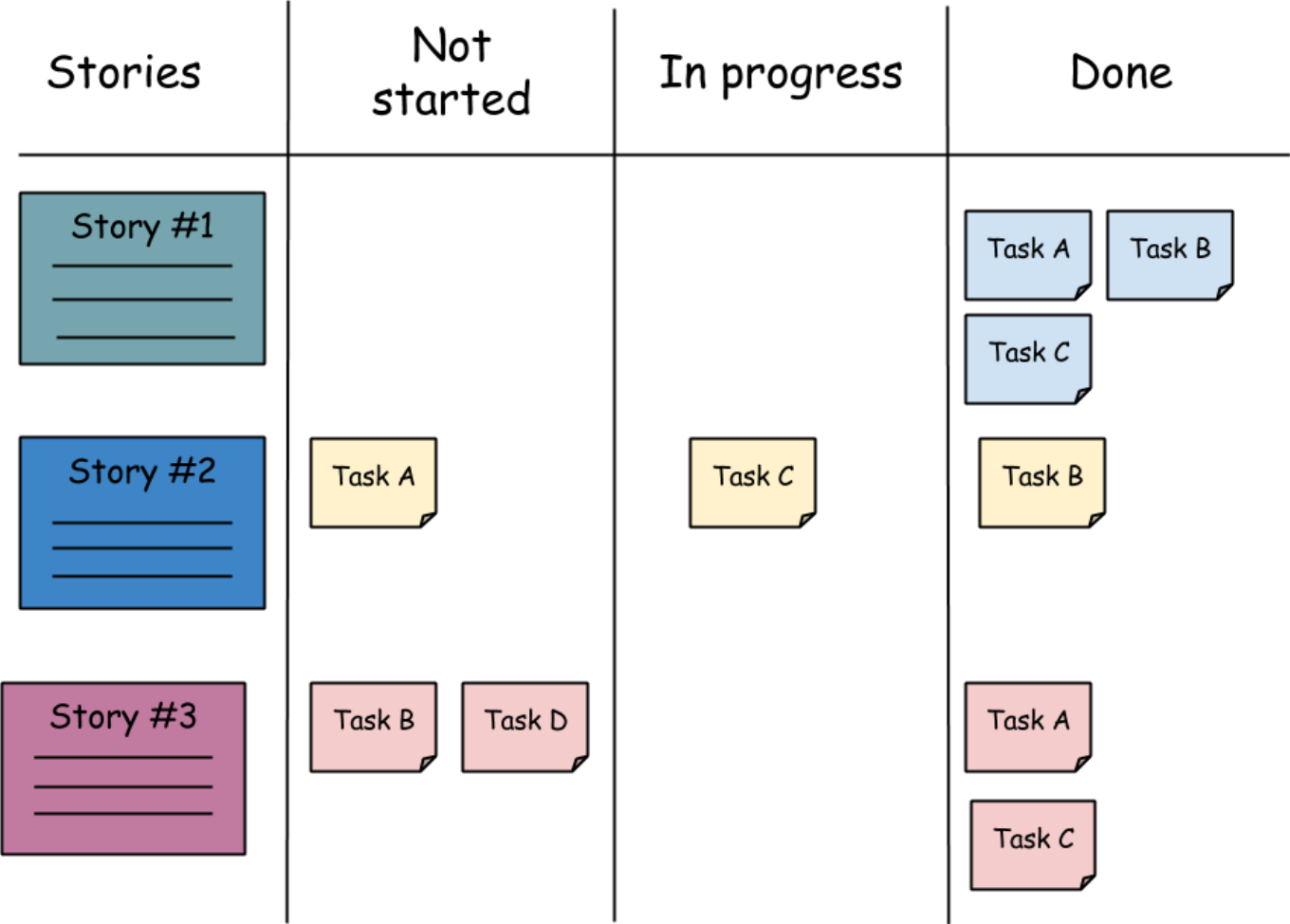
ACCEPTANCE CRITERIA

- In agile, **acceptance criteria** refer to a set of predefined requirements that must be met to mark a user story complete

- A product owner may be responsible for writing acceptance criteria for the stories in the product backlog
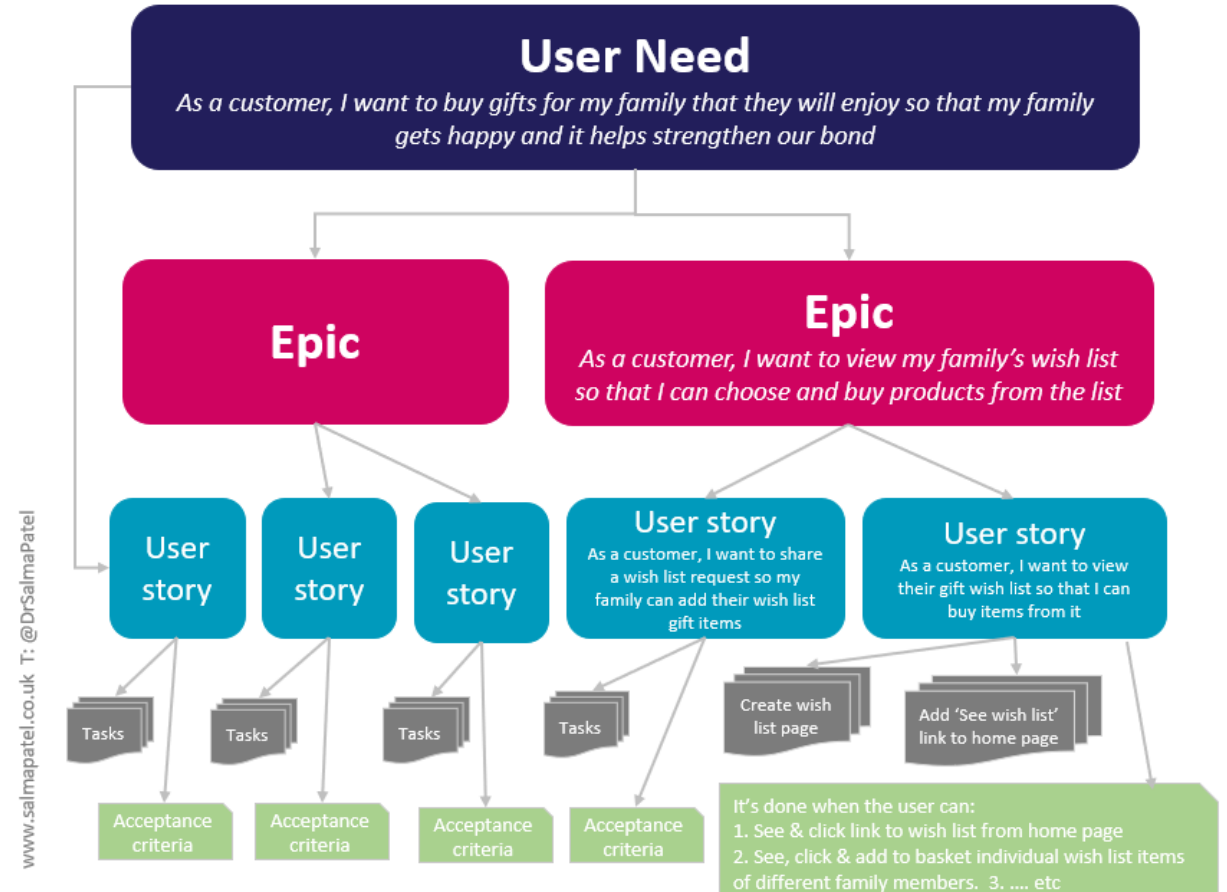
SCENARIO-BASED ACCEPTANCE CRITERIA

Also known as the Given/When/Then (GWT) acceptance criteria, with 5 components:

- Scenario: A description of the situation that the user will encounter.
- Given: The starting state of the scenario.
- When: The action that a user takes.
- Then: The result of the action in the previous "When".
- And: used to continue any of three previous statements

SCENARIO-BASED ACCEPTANCE CRITERIA

GIVEN / WHEN / THEN

User Story — As a user, I want to be able to continue an episode from where I left off after closing the app or watching another show.

Scenario — The user wants to continue the episode and therefore goes out of their way to search for the show.

Given — User searches for their missing episode manually.

When — The user plays the episode.

Then — The episode plays back at the exact timestamp that they previously reached.

ACCEPTANCE CRITERIA
NETFLIX EXAMPLE 01

https://userwell.com/acceptance-criteria-types-and-tips/#h-1-given-when-then

SCENARIO-BASED ACCEPTANCE CRITERIA

https://userwell.com/acceptance-criteria-types-and-tips/#h-1-given-when-then

SCENARIO-BASED ACCEPTANCE CRITERIA

User story: As a website user, I want to be able to recover the password to my account, so that I can access my account in case I forget the password

Scenario	Forgot password
Given	The user navigates to the login page
When	The user selects *forgot password* option
And	Enters a valid email to receive a link for password recovery
Then	The system sends the link to the entered email
Given	The user receives the link via the email
When	The user navigates through the link received in the email
Then	The system enables the user to set a new password

https://www.altexsoft.com/blog/acceptance-criteria-purposes-formats-and-best-practices/

SCENARIO-BASED ACCEPTANCE CRITERIA

User story: As a bank card user, I want to be able to request cash from my account at an ATM so that I will be able to receive the money from my account quickly and in different places.

Scenario 1	Requesting the cash from a creditworthy account
Given	The account is creditworthy
And	The card is valid
And	The dispenser contains cash
When	The customer requests cash
Then	Ensure the account is debited
And	Ensure cash is dispensed
And	Ensure the card is returned

https://www.altexsoft.com/blog/acceptance-criteria-purposes-formats-and-best-practices/

RULE-BASED ACCEPTANCE CRITERIA

Also known as Verification List, which consists of a checklist of pass/fail statements that need completion to be successful.

User Story: As a user, I want to register for an account at a website.

- The "Register" button on the homepage takes the user to a registration page.
- The user can input their desired username.
- User can input their desired password.
- The user has to retype their password in a verification box.
- If the two passwords do not match, the user will see a warning message.
- For their password, the user has to use certain sets of characters (e.g. uppercase, lowercase, numbers, symbols) as defined by the security standards set by the product specification.
- If the user lacks these characters in their password, a warning indicates what they're missing.
- The user can't type more than 22 characters in their password.
- The user can input their email address.
- After registering, the user receives a verification email.
- The user can input their verification details in a field.
- If the input is wrong, the user sees a warning message.
- The user can resend their verification email.

THE IMPORTANCE OF ACCEPTANCE CRITERIA

- **Narrowing down the scope of the product**, precisely describing how to fulfill user stories.

- **Easier to test the product**. Acceptance criteria have clearly defined conditions for success and failure, which can be easily tested during QA.

- **Addressing invalid or negative outcomes**. User stories only describe happy users who have fulfilled expectations of the product functionality. Acceptance criteria, on the other hand, can show alternative, non-successful user paths that must be accounted for.

MILESTONE 1 - REQUIREMENTS

Blackboard -> Assignments -> Project Proposal

Please read the documentation carefully for your first milestone!
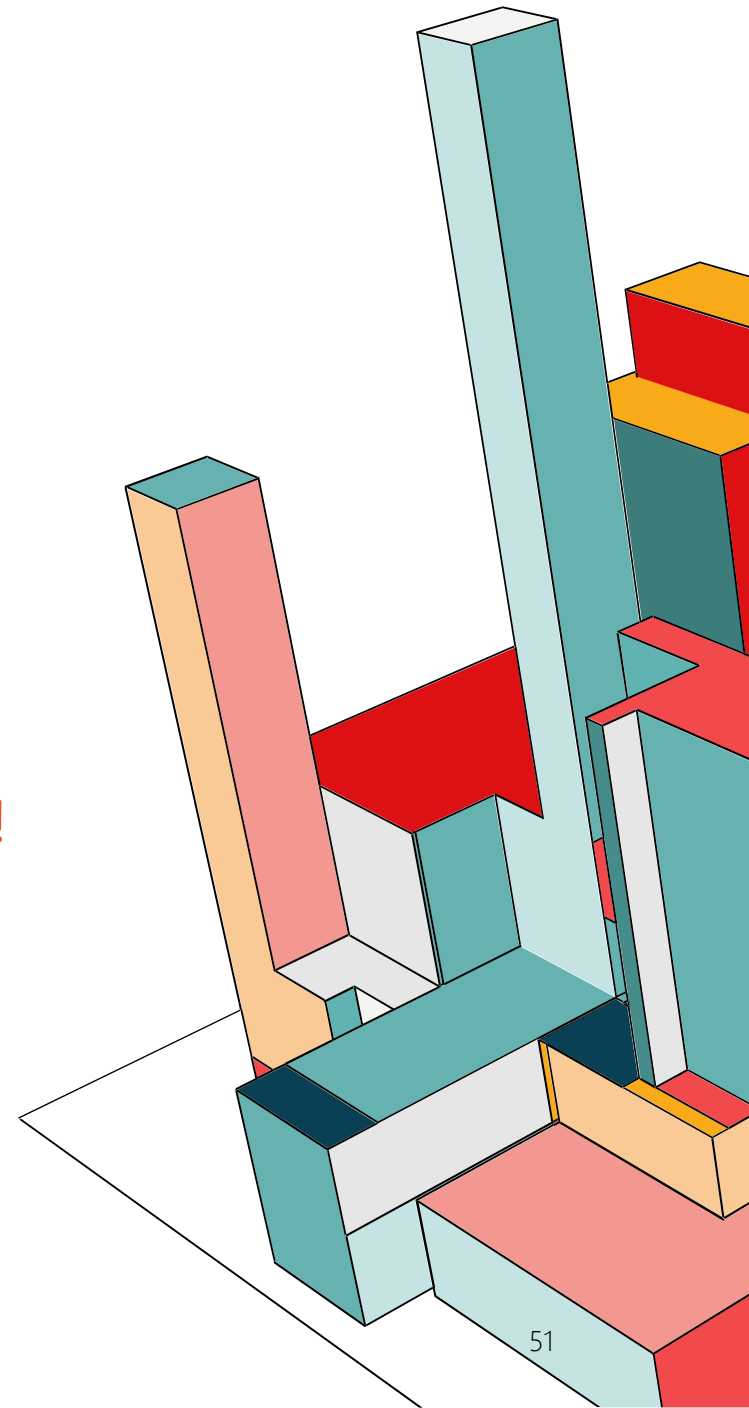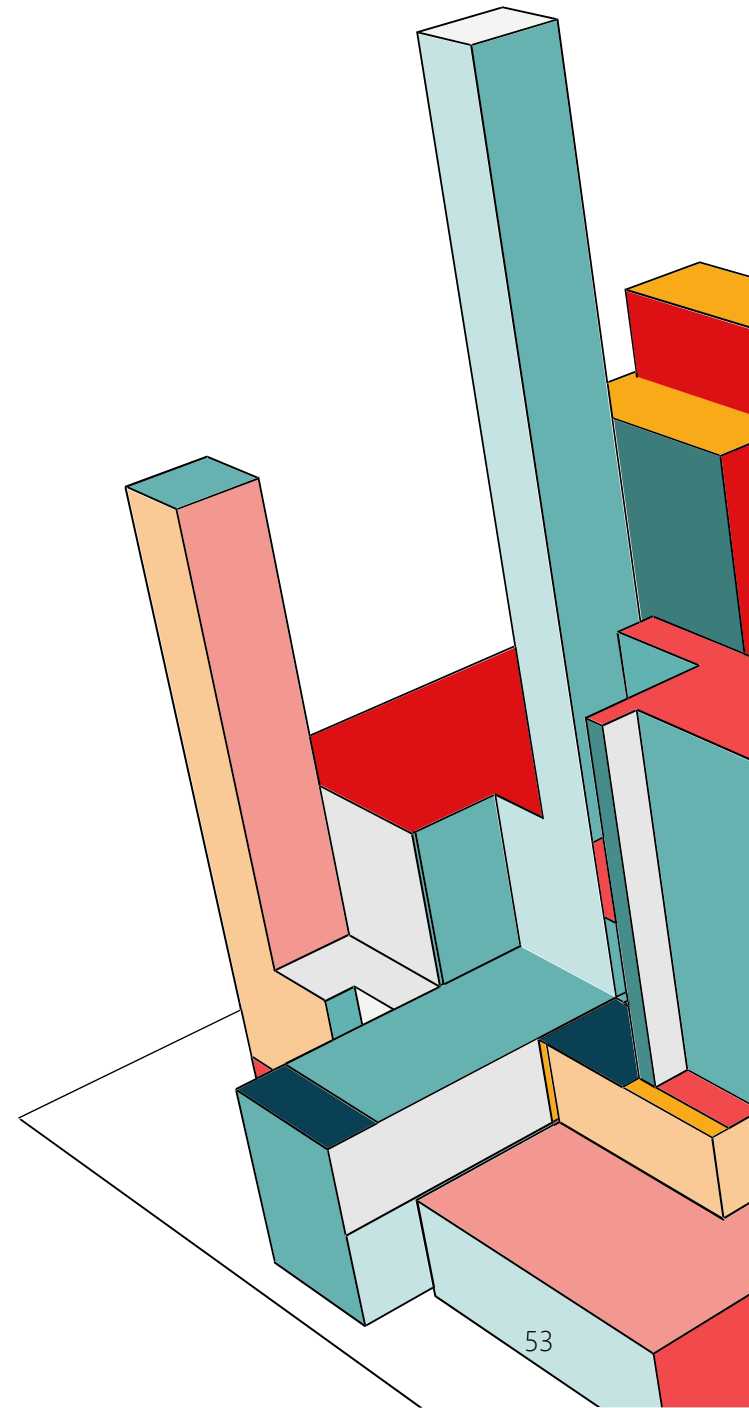
A NOTE ON UML

- The Unified Modeling Language (UML) is a general-purpose, developmental modeling language in the field of software engineering that is intended to provide a standard way to **model the requirements and design of a system**.

- UML diagrams are heavily-associated with the Waterfall process (Big upfront design), but may have limited usage in processes like Scrum that focus on continuous changes.

READINGS

- Chapter 8-11. Software Engineering A Practitioner's Approach by Roger Pressman, 8th edition.

- Chapter 5. Software Engineering by Ian Sommerville. 10th edition

- 第8章 软件需求. 现代软件工程基础 by 彭鑫 et al.

NEXT

- Version Control System