SOFTWARE DEVELOPMENT PROCESS

LECTURE 5:
USER STORIES AND AGILE ESTIMATION

Asst. Prof. Dr. ISARA ANANTAVRASILP

USER STORIES

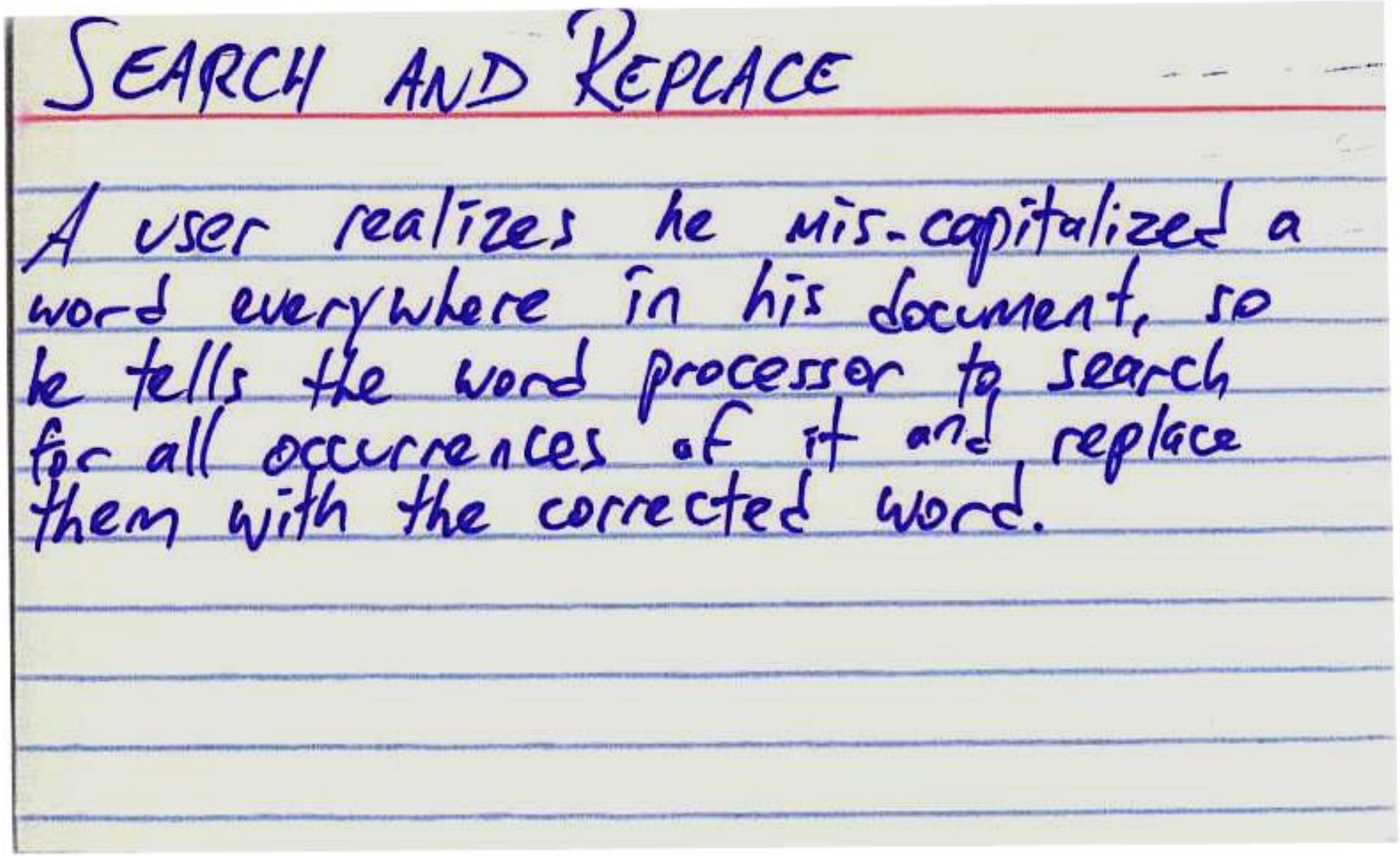
INVOLVING **USERS**

- Agile approach emphasizes on user involvement
  - They are the ones who know their domain best
  - They know what they want and they know what they want first
  - **The user should involve thorough the project**
- So, the users (and other stakeholders) should involve in system requirement elicitation
  - But they do not know modeling (e.g. use cases)
  - They do know how to come up with functional / non-functional requirements
- Then, what should we do to let the users express their needs, requirements and goal?
  - **User stories**

USER **STORIES**

- **User story**: A short, simple description of a feature told from perspective of a stakeholder
- It is usually written on an index card or sticky note:
  - As a [*type of user*], I want [*some goal*] so that [*some reason*]
  - As a ***astronaut***, I want a ***align my ship*** so that I can ***park at the space station***

USER STORIES **EXAMPLE**



SEARCH AND REPLACE

A user realizes he mis-capitalized a word everywhere in his document, so he tells the word processor to search for all occurrences of it and replace them with the corrected word.

Image: http://www.stellman-greene.com/2009/05/03/requirements-101-user-stories-vs-use-cases/

USER STORIES **EXAMPLE**

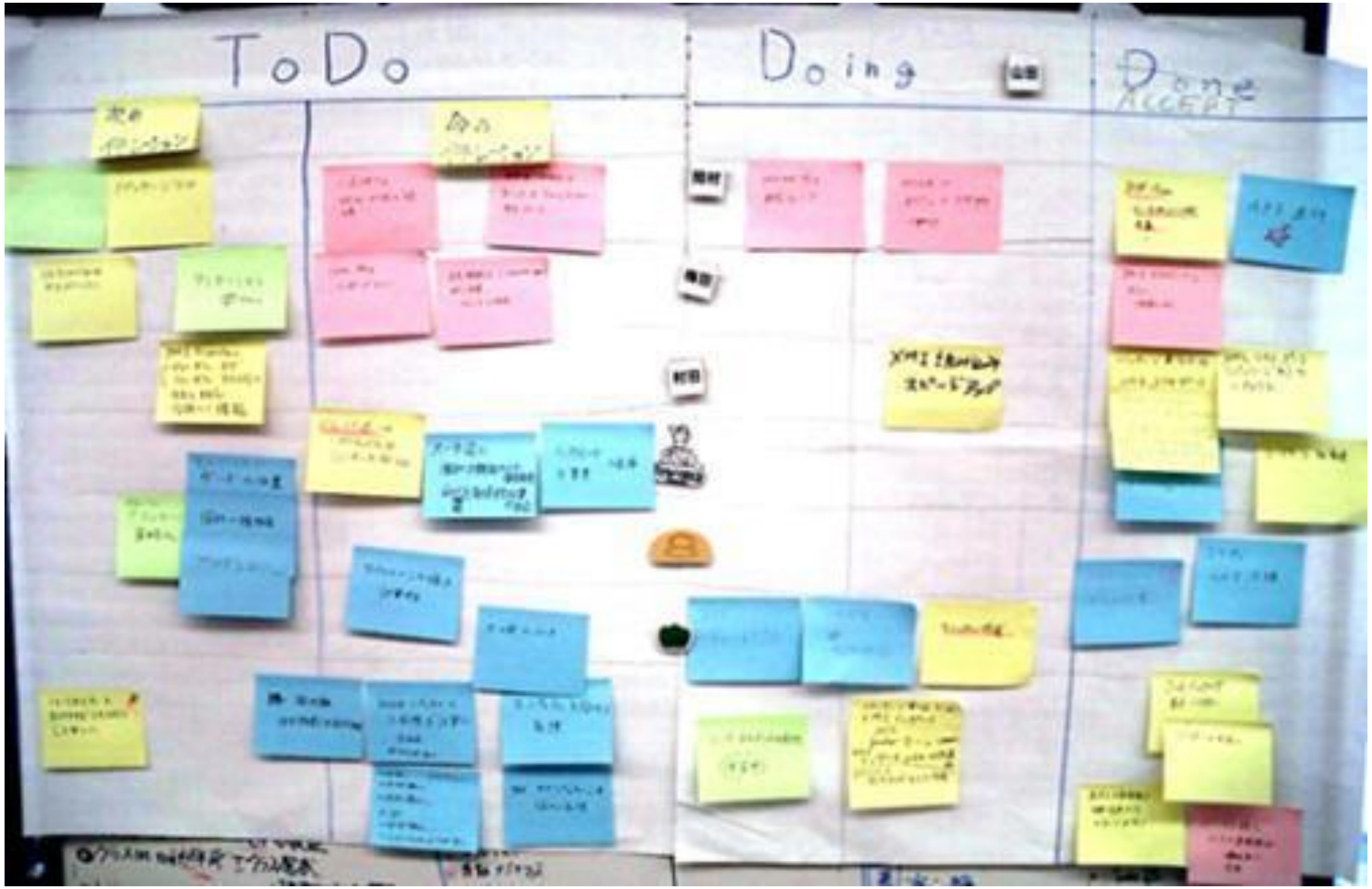


Image: http://www.agilebuddha.com/agile/story-mapping-andvs-process-maps/

USER STORIES VS USE CASES VS SCENARIOS

- A user story is an informal, user-oriented description of use case
  - Both user story and use case describe interaction between the actors and the system
  - But many stories may correspond to only one use case and one story may span into many use cases
  - User story is minimal, telling the user’s needs
  - Use case is complete, explaining what the system must do to serve the user’s needs
- Scenario is a story of actor(s) interacting with a system
  - The interactions are laid out in sequence
  - One scenario may consist of many use cases

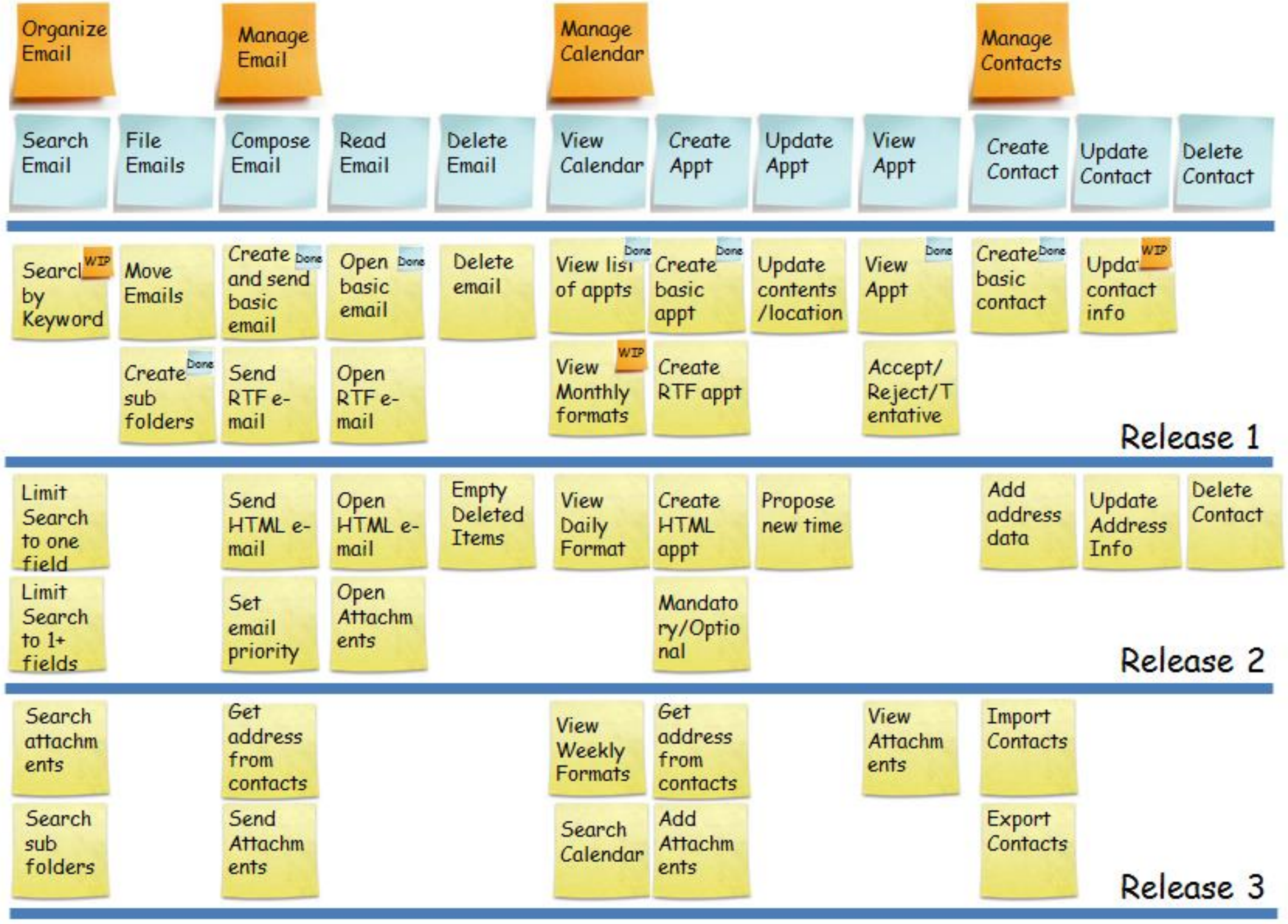
EMERGENT **REQUIREMENTS**

- **Emergent Requirements**: Features or functionalities that are not anticipated before
  - **Problems** that are discovered during the project
  - **New** ideas
  - Required features that everyone has **overlooked**
  - These issues can be seen only after you use your software
- Emergent requirements can always exist
  - Non-trivial projects: You have no clear idea in the application domain or problem
  - Research and innovative projects
- Such requirements make it impossible to plan the project
- Agile approach emphasizes on working code after each iteration to detect emergent requirements early

EPICS

- To deal with emergent requirements:
  - Assume that they exist, and we cannot think of everything
  - Do not try produce backlog that contains all user stories that may happen
- Instead, think about big, **epic** stories first
- **Epics**: A large user story that must be broken down into small ones before putting into an iteration
  - Example: “Land on the moon” “Build an elevator”
- We sort the epics first then break it down into smaller stories or functionalities when we actually implementing them

USER STORIES CAN BE GROUPED AND PRIORITIZED



Organize Email		Manage Email			Manage Calendar				Manage Contacts			
Search Email	File Emails	Compose Email	Read Email	Delete Email	View Calendar	Create Appt	Update Appt	View Appt	Create Contact	Update Contact	Delete Contact	
Search by Keyword WIP	Move Emails	Create and send basic email Done	Open basic email Done	Delete email	View list of appts Done	Create basic appt Done	Update contents /location	View Appt Done	Create basic contact Done	Update contact info WIP		
	Create sub folders Done	Send RTF e-mail	Open RTF e-mail		View Monthly formats WIP	Create RTF appt		Accept/Reject/Tentative				Release 1
Limit Search to one field		Send HTML e-mail	Open HTML e-mail	Empty Deleted Items	View Daily Format	Create HTML appt	Propose new time		Add address data	Update Address Info	Delete Contact	
Limit Search to 1+ fields		Set email priority	Open Attachments			Mandatory/Optional						Release 2
Search attachments		Get address from contacts			View Weekly Formats	Get address from contacts		View Attachments	Import Contacts			
Search sub folders		Send Attachments			Search Calendar	Add Attachments			Export Contacts			Release 3

Image: http://www.agilebuddha.com/agile/story-mapping-andvs-process-maps/

AGILE ESTIMATION

AGILE ESTIMATION

- **Agile estimation** is a software estimation approach that is based on the principles of Agile software development.
- It involves breaking down a software project into smaller, manageable parts and using various techniques to estimate the effort required to complete each part.
- The goal of Agile estimation is to provide a flexible and adaptive approach to software estimation that can handle the uncertainty and unpredictability of software development projects.

AGILE ESTIMATION

- The focus is on developing a rough estimate that is based on the team's understanding of the work, rather than a precise calculation. This allows the team to be flexible and adapt to changes as the project progresses.

- Agile estimation also emphasizes the importance of collaboration and communication between team members. The estimation process is often done as a team. Everyone is encouraged to share their expertise and knowledge. This helps to ensure that the estimate is based on a shared understanding of the work and reduces the risk of misunderstandings or missed requirements.

AGILE ESTIMATION **METHODS**

- **Planning Poker**: This is a consensus-based estimation technique that uses a deck of cards with numbers to estimate the effort required for a particular task or feature. Team members choose a card that represents their estimate and then discuss their reasoning until a consensus is reached.

- **T-Shirt Sizing**: This is a relative estimation technique that uses t-shirt sizes (such as S, M, L, XL) to estimate the effort required for a particular task or feature. Team members can quickly agree on a relative size and waste no time in specific numbers.

- **Wideband Delphi**: This is an estimation technique that involves a group of experts making independent estimates of a task or feature, followed by a facilitated discussion to reach a consensus. The estimates are anonymous, and the facilitator provides feedback to the group to help them converge on a consensus estimate.

AGILE ESTIMATION **METHODS**

- **Affinity Estimating**: This technique involves grouping similar tasks or features together and then estimating them as a group. This can be a quick and efficient way to estimate a large number of items, without getting bogged down in the details of each one.

- **Dot Voting**: This is a simple technique that involves team members placing dots next to items they feel are the highest priority. This can help to prioritize work and make sure that the team is working on the most important items first.

ANCHORING BIAS

- Anchoring Bias: A cognitive bias that describes the tendency for people to rely too heavily on the first piece of information they receive when making decisions or estimating values.
- **For example:** The first developer says, “I think this story takes 30 days.” The other team members would immediately estimate relating to 30 days instead of having an unbiased estimation.
- This is also true when you ask for certain salary.
- This initial piece of information is known as the anchor and can have a strong influence on subsequent decision making.

ANCHORING BIAS

- Anchoring bias is especially prevalent when there are no other references or other support information.
- It is hard to be unbiased in a discussion because once someone estimate a value, others cannot help but to think on that value.
- Agile estimation methods, therefore, try to avoid the bias by forcing to produce the estimation at the same time.

PLANNING POKER

- **Planning Poker**: A process to estimate effort to complete a user story that uses a deck of cards with numbers to estimate the effort required for a particular task or feature
- It is a consensus-based method where the team member must estimate together and agree upon the result
- In the estimation process, no one is allowed to speak to avoid anchoring
- Once the estimations of all members are revealed, the team can start discussing their reasoning

PLANNING POKER **CARDS**

- A set of cards, usually showing a Fibonacci sequence
  - The larger the estimation, the larger the uncertainty
  - That is, you cannot say “17 days”, it is too precise for such large estimation. You can only guess widely.
- The number can represent any work units e.g., days, hours, half a day

Images: http://en.wikipedia.org/wiki/Planning_poker http://www.it-zynergy.com/scrum-planning-pokers

PLANNING POKER **RULES**

- To estimate a story, the moderator selects a story and let the team members (may include PO and other stakeholders too) ask and discuss about the tasks
- Then, each team member select a playing card, **face down**, based on estimated effort. The player must not speak the number out loud
- Once everyone has selected a card, all member reveal their cards
- The players with the highest and lowest numbers are allowed to state their reasons. The others can later join the discussion
- Repeat the process until a consensus is reached
- All discussions must be time-boxed e.g. 3 minutes

T-SHIRT SIZING

- **T-Shirt Sizing**: A quick and simple way to get a rough idea of the effort required using only few relative estimated sizes.
- The technique gets its name from the use of t-shirt sizes to represent different levels of effort.
- Typically, teams will use t-shirt sizes like Small, Medium, Large, Extra Large, and so on. These sizes are used to represent different levels of effort, rather than precise measurements of time.

T-SHIRT SIZING **CARDS**

- A set of cards showing “t-shirt” sizes
- The team can set any sizes and agree upon the definition of each t-shirt size
- For example,
  - Small: 1-2 man-days,
  - Medium: 3-4 man-days
  - Large: 5-6 man-days.
  - XL: 7+ man-days.
  - XXL: An epic

XS
S
M
L
XL
X XL
∞
Too Complex
?

https://teamworx.co.nz/agile-articles/print-planning-poker-cards-fibonacci-t-shirt-sizes/

T-SHIRT SIZING **RULES**

- Identify the items to be estimated: This could be a list of features or a set of user stories that need to be completed

- The PO explain the story so that everyone understands what to do. The team can ask questions if there are any unclarity

- Estimate each item: The team looks at each item and decides which t-shirt size best represents the level of effort required.

- Review and discuss the estimates: The team discusses their estimates and decides if they are happy with the results. If there are any large disparities in the estimates, the team can discuss why and adjust the estimates accordingly.

ProTip

IF YOU ARE ALWAYS RIGHT, YOU ARE WRONG!

- Numbers in planning poker represent “some” unit.
- It could be just a number to present how large a task is comparing to *other* tasks.
- It could also be absolute number or days or weeks.
- But you cannot always be right.