Software Engineering

University Computer Science · Vocabulary in context

Word	Example
software engineering	**Software engineering** covers designing, testing and maintaining software.
requirements engineering	**Requirements engineering** finds out what users really need.
functional requirement	“Users can log in” is a **functional requirement**. “Users can log in”
non-functional requirement	“Pages load in under one second” is a **non-functional requirement**. “Pages load in under one second”
use case	A **use case** describes how a user reaches one goal.
layers /ˈleɪəz/	The app has three **layers**: screen, logic and storage.
dependency rule	The **dependency rule** says the core code must not depend on the screen code.
cohesion /kəʊˈhiːʒn/	High **cohesion** means a class does one clear job.
coupling	Low **coupling** means classes depend on each other very little.
states /steɪts/	A traffic light has three **states**: red, amber and green.
transitions /trænˈsɪʃnz/	The **transitions** show how the system moves from one state to another.
branching model	A **branching model** sets rules for using branches in Git. Git

exercises.lol exercises.com.cn

Word	Example
merge conflicts	Fix **merge conflicts** when two people change the same line.
test pyramid	The **test pyramid** has many unit tests at the bottom and few end-to-end tests at the top.
user stories	**User stories** describe what a user wants, such as “I want to reset my password”. “I want to reset my password”。
standup	At the daily **standup**, each person says what they did and what is blocking them.
retro	At the **retro**, the team discusses how to work better.
threat model	A **threat model** lists possible attacks and how to stop them.