16L-4059 (NS-B)
Raphael

Q. 2(a) Secure Software Development Lifecycle

Requirements Engineering

1) Requirement Elicitation:	Gathering requirements is important because it gives direction for whole software project.
2) Define Functional Requirements via use cases etc.	: Useful in specifying software modules & testing in later stages.
3) Define security features & requirements	: Defining security, according to required software project, has benefits for later stages
4) Define misuse, abuse cases &/or attack trees	: Similar benefits as (2) FRs.

Analysis Phase:

5) Risk Analysis:	Useful in saving time later on, by investing more resources to high-risk items.
6) Expand Sec. Req. to multiple goals, constraints & tasks	Useful in design & security inspection phase.
7) Select security mechanisms.	Necessary to satisfy security requirements
8) Specify FRs for security mechanisms	Extension of previous step/activity.
9) Assess security index.	If measure not acceptable, repeat (6) - (8) activities.
10) Review inspections of all requirements	To detect any missed out or added risks.

16L-4059 (NS-B)
Raphael

Requirements Phase	
11) Prioritize all requirements	Requirements are not all equally important or equally threatening. Saves time later on.
Design Phase	
12) Construct design diagrams	Improves implementation
13) Inspections (design)	Early detection of vulnerabilities
14) Remove vulnerabilities	Obvious extension of previous
15) Assess security index & coherence	Repeat 12-14 if new issues detected.
16) Embed security monitor	Useful in testing & maintenance phase.
Implementation Phase	
17) Select secure programming language(s)	Reduce chances of unexpected future attacks.
18) Follow secure coding standards & guidelines	Tried & tested standards offer greater benefits of protection.
Testing Phase	
19) Unit Testing	Basic & smallest form of testing.
20) Functional Testing (Blackbox/whitebox)	Broader form of testing needed for larger software modules.
21) Integration Testing	Necessary before deployment.
22) Security: Pen testing, static & dynamic analysis etc.	Necessary to test security.
Maintenance Phase	
23) Monitor software behavior	Necessary to ensure maximum safety likelihood.
24) Locate vulnerabilities	Extension of (23) if issues found.
25) Resolve & remove	Extension of (24).

16L-4059 (NB-B)
Raphael

Q.2(b) Removed activities are listed below:
(Numbering according to previous ~~part~~)

3) Define security	Reduce time by simply writing security requirements. Skip discussion & definitions.
6) Expand security reqs. to goals ...	~~Skip completely~~ Decide security mechanisms from requirements directly; defining requirements is not extremely important.
10) Review inspections of all requirements	Go directly to (11) "Prioritize"; chances are that extreme vulnerabilities don't exist at all.
15) Assess SI & repeat activities	Only repeat 12-14 for 1-2 iterations. Not loop over until perfect SI reached.
18) Coding standards	Save time by coding faster (rely on personal experience).
22) Security testing: Pentesting, static & dynamic analysis	To save time, ~~skip one of~~ do not employ all types of testing. Employ reduced set of testing techniques.
23-25 Maintenance Activities	According to remaining budget, trim off these activities. Notify upper management of ~~extent~~ lack of maintenance due to budget.