

4156

9/27/16

two assignments coming up

due this Thursday - team initial proposal  
meet with ANY IA to discuss  
before submitting

will hold team meetings during class  
on Thursday, a Ewan will be  
here for the JP Morgan teams

team mentors for on-campus non-JP Morgan  
teams determined after you tell  
us (in proposal) what tech you plan

due next Tuesday - pair practice with tech  
build toy system that uses your tech  
to make sure it does what you think  
since can change for revised proposal  
if not sure, each pair in team might  
try \* different \* tech

---

finish pair programming

4156

9/27/16

Some requirements cannot easily be expressed as features (User stories or use cases)

called non-functional requirements (NFRs)

possibly better thought of as constraints on the design & implementation

most applications, if not all, have some NFRs that are just as important to success as functional requirements (features)

in this course our main NFRs are reliability & security

other NFRs -	performance
quality	scalability / availability
attributes	usability / accessibility
"ilities"	maintainability
	manageability
	portability
	interoperability
	regulatory
	...

need to include NFRs in "definition of done"

4156

9/27/16

reliability = degree to which sw produces  
stable & consistent results

generally, means sw should always  
produce same output for same inputs

considering all inputs & outputs

e.g. application state

files & databases

user/network I/O

system/library calls

also doesn't crash

doesn't lose or corrupt user data

recoverable when hardware crashes

this is NOT the same thing as no bugs  
no non-trivial software has zero bugs  
instead means no show-stopper bugs  
(can't be worked with or worked around)

we will assume all software must be  
reliable, whether customer asks for  
it or not



4156

9/27/16

security - show mages

authentication, integrity & availability

authentication - prevent unauthorized users  
for accessing data or performing  
application functions

includes non-repudiation (audit trail)

integrity means user data can not be  
changed or deleted  
(different from corruption issue)

availability - freedom from  
denial of service  
(not same as scalability issue)

more about security later in class  
reliability permeates class -  
best practices  
testing  
code inspection

for now, we consider only how thrs  
fit into iteration planning

4156

9/27/16

we could write a user story like

"as a user, I want this software to be reliable and secure"

but that doesn't help - too generic

so instead we will think about both cross-cutting & feature-level risks, error situations & threat models

cross-cutting issues should be phrased as user stories

feature-level issues should be included as part of that feature

→ all user stories should implicitly include certain elements

e.g. only the specified role(s)

can run this operation

all user input must be sanitized

all database, files, user & network output must be logged

all sensitive operations must be logged

all file operations should record / check checksums