UNIT 6 REVIEW

http://www.flickr.com/photos/tropicaliving/3672347622/sizes/o/

Key concepts in this unit

- Agile manifesto: individuals & interactions, working software, customer collaboration, responding to change
- DSDM vs XP
- User stories, tasks, units
- Pair programming
- Agile modeling / diagrams

Check yourself

- Which of the following processes is most likely to have 1 week iterative cycles?
  A. DSDM
  B. Scrum
  C. XP
  D. Waterfall with prototyping

Check yourself

- Which of the following processes is most likely to have facilitated workshops?
  A. DSDM
  B. Scrum
  C. XP
  D. Waterfall with prototyping

Check yourself

- Which of the following processes is most likely to have the most lengthy documentation of requirements?
  A. DSDM
  B. Scrum
  C. XP
  D. Waterfall with prototyping

Check yourself

- How are requirements documented in XP?
  A. Paper prototypes
  B. Use cases
  C. User stories
  D. UML class diagrams

Check yourself

- When are the engineers allowed to prioritize user stories in XP?
  A. Always
  B. If the customer lacks software engineering skills
  C. If the customer pays bills late
  D. Never

Check yourself

- What is the primary benefit of a spike?
  - A. By trying a simple version of a task, a spike reveals how hard the real task will be.
  - B. By deleting requirements from the project, a spike reduces the effort required.
  - C. By imitating the way a customer will test code, a spike increases the system quality.
  - D. None of the above

Check yourself

- In XP, software engineers should…
  A. Be heroes: work all night if needed, and on weekends!
  B. Keep the customer happy at all costs, even if it means lying about project status.
  C. Be courageous, willing to always do and say what is right even when it is not popular.
  D. None of the above

Check yourself

- In XP pair programming, the “pilot” should *NOT* do the following…
  A. Worry about the screen font (the co-pilot can always scoot closer to the screen)
  B. Allow the co-pilot to interrupt with questions
  C. Use APIs or tools that are new to the co-pilot
  D. Write code for the project until after unit tests have been started

Check yourself

- When is it acceptable to create diagrams / models in agile processes?
  A. When communicating with customers
  B. When communicating with other programmers
  C. When trying to think something through
  D. All of the above

Check yourself

- Which statement is true?
  A. Waterfall has requirements; agile does not.
  B. Waterfall involves diagrams; agile does not.
  C. Waterfall involves testing; agile does not.
  D. Waterfall always includes formal analysis; agile does not.
  E. Waterfall rarely includes paper prototypes; agile almost always includes paper prototypes.
  F. None of the above.