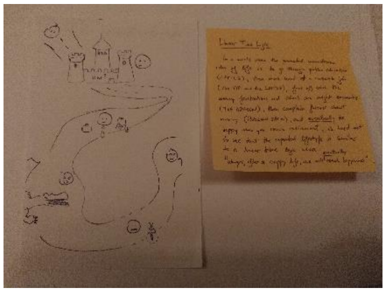
- **SSVV 2019-2020**
- **Take Home Exam Subject – Individual tasks**
- **300 XP**
- **The answers for each question must be written by hand!**
- **On each submitted solution, please write your name and group.**
- Short 4 questions
  - 100 XP
  - Provide the definition, explain the concept, answer the question.
  - Maximum 1 A4 page
    - short description of the notion - a paragraph (5-8 lines)
  - Question 1: White box testing – predicate coverage (definition +example)
  - Question 2: Symbolic execution tree (definition + example)
  - Question 3: Model checking - liveness properties (definition +example)
  - Question 4: Dijkstra weakest preconditions (definition + example)
- Long question – choose one from the list.
  - 100 XP
  - Maximum 1 A4 page
  - Compare and contrast 2 concepts.
    - Concept 1 = definition + simple example
    - Concept 2 = definition + simple example
    - Provide 4 similarities/differences of the 2 concepts.
  - Choose one from the following list.
    - Compare and contrast integration testing versus unit testing.
    - Compare and contrast Smoke testing and Load testing.
    - Compare and contrast Inspection and Testing.
    - Compare and contrast Testing and Pair-programming
    - Compare and contrast: test automation and manual testing.
    - Compare and contrast: Symbolic execution and Testing
    - Compare and contrast: Testing and Model checking
    - Compare and contrast: Quality assurance vs Quality control
    - Compare and contrast: Functional and non-functional testing types
    - Compare and contrast Model checking and Demonstrating correctness
    - Compare and contrast Testing and Demonstrating correctness
    - Compare and contrast: top-down integration and bottom-up integration
- Fun assessment regarding SSVV
  - 100 XP
    - 50XP



      - Create/draw with colors pencils a DIXIT card with notion from SSVV.
      - Explain in a paragraph how someone may use the card to play (the subtle SSVV notions embedded in the card).
    - 50 XP
    - Draw one scenario from the list below.
      - Draw a reason why a bug might slip to production.
      - Draw a reason why do bugs occur.
      - Draw a tester and a developer having a conversation about testing/bug/etc.
      - Draw the nastiest (software) bug you ever found. Explain how did you discover it.
      - Draw a picture to illustrate: "Why should we "fix" bugs asap."