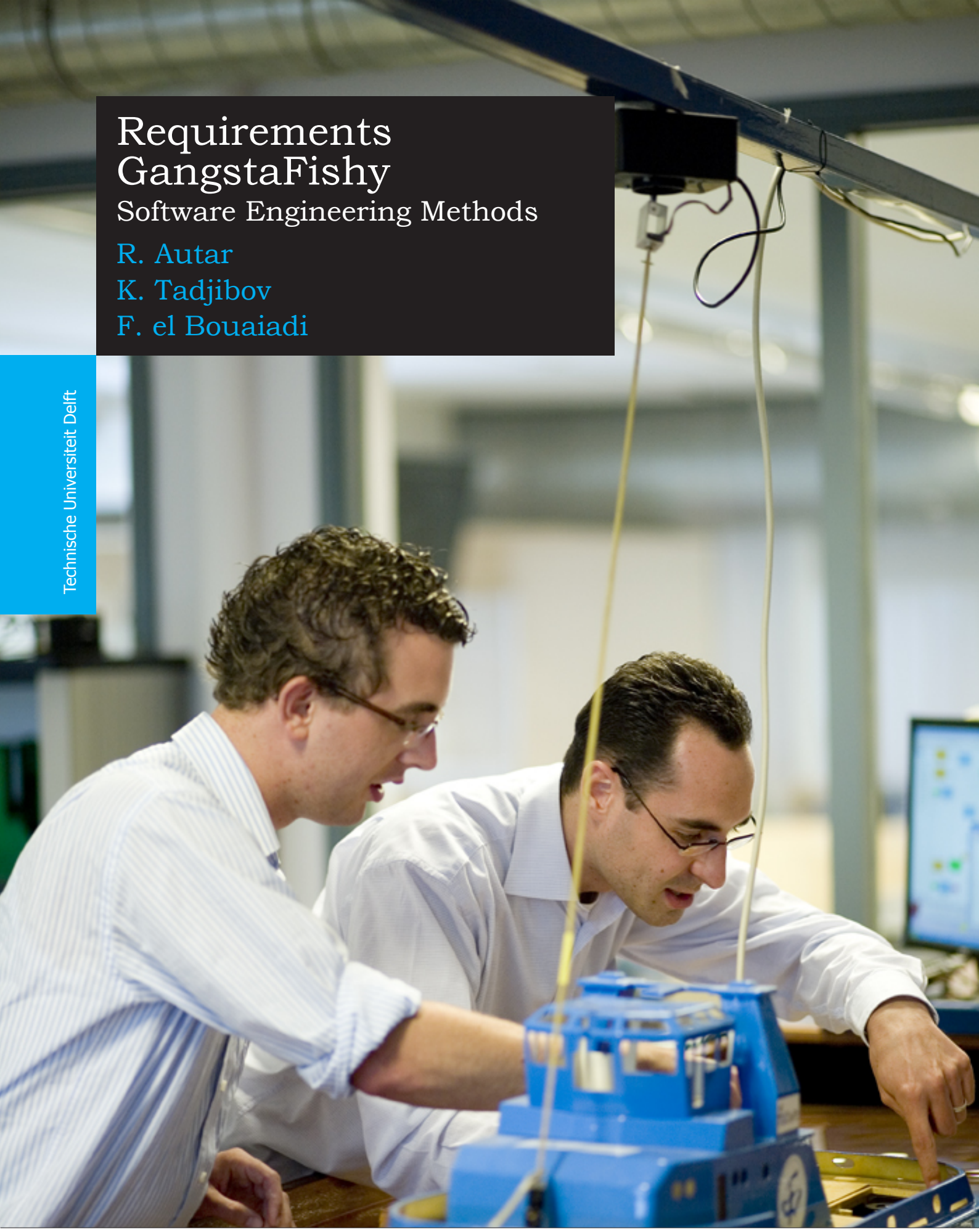
Requirements GangstaFishy

Software Engineering Methods

R. Autar
K. Tadjibov
F. el Bouaiadi



Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Contents

1

Functional Requirements

For the game GansgtaFishy, the functional requirements can be grouped and prioritized in the MoSCoW model. In the following subsections the functional requirements

1.1. Must Haves

- At the beginning of the game you have a start pane with the following options:
  - Start
  - Instructions
  - About
- If the play button is pressed, the game will start.
- At the beginning of the game the player should be able to move a small fish with the buttons up, down, left and right.
- Enemy fishes will continue appearing that will be smaller, equally big or bigger than player's fish.
- If the player moves his fish towards enemy fish that is smaller or equally big as his fish, the enemy fish will be considered eaten.
- If the player's fish eats a enemy fish, the player his score will increase. Also the size of the player's fish will increase.
- If the player's fish comes in contact with a fish larger, the player's fish will be considered eaten and the game will be ended.
- At a certain extent the player's fish will stop growing, however the game will continue till the player's fish has been eaten.
- When the player's fish is eaten and the game is over, the player gets sent back to the start screen.
- If the player gets a higher score than the previous tries, then the previous score will be overwritten. Else the previous score is kept.

1.2. Should Haves

- The player should be able to turn the sound of the game on and off during the game or right from the beginning (at the start screen).
- The player should be able to pause and resume the game.
- The enemy fishes should have different colors and shapes indicating different types of fish.
- If the player's fish moves to the end of the gaming screen, he should appear back on the other end of the screen i.e. :
  - If he moves towards the right end of the screen, he should reappear on the left end of the screen.
  - If he moves towards the upper end of the screen, he should reappear on the down end of the screen.
- The fish should have movement inertia i.e. it should take some time to move from complete stillness and also take time to stop when already moving respective to the current speed.
- The different enemies should have random speeds at which they move.
- The different enemies should appear from different places in the gaming screen.
- The keys W, A, S and D should also be able to move the player up, down, left and right respectively.

1.3. Could haves

- If the player pauses the game, the player gets a menu with options: settings, instructions and quit game.
- If the players score higher that the top 5 scores previous scores, the player can fill in his/her name in the high score table.
- The player can save/load the game.
- The player's fish can call for backup of piranha to eat more/larger fish.
- The player's fish can carry a gun.
- The player can change the controls for the fish to move.
- The screen will also display how many enemy fishes the player's fish still needs to eat to get bigger.
- The player can share his/her score on facebook and twitter.

- The fish has collision boundaries, that perfectly fits the sprite.
- Some enemy fish could give the player's fish power-ups, such as extra life or grow the player's fish for a period of time.
- Some random pick-up items could appear at random places to let the player's fish grow or give extra life.
- Some pick-up items might lead the fish to slow down or get smaller or make enemy fishes less visible.
- Some pickups may lead for the player's fish to teleport to a different lake with different enemies.
- The player may choose what fish he want to play as before the game starts.

1.4. Would Haves

- The game has an option multiplayer.
- The score of the game can be shared on linkedin
- The player can customize his own fish.

2

Non-Functional Requirements

Unlike functional requirements, following requirements do not describe what the game should do, but instead describe what conditions had to be met during the development of the game.

- The game shall be playable on Windows (7 or higher), Mac OS X (10.8 and higher), and Linux.
- The game shall be implemented in Java.
- A full working version of the game should be delivered due 11 September.
- The game and its classes shall be well documented.
- During the development process, after the first working version has been delivered, the SCRUM method will be used for code iteration.
- The code of the game shall have at least 75 percent of test coverage.