From the office of Quentin Raffles at Watson Games

Some further specification updates for you:

Q) Are tokens allocated to players randomly or can players choose?
A) It would be nice to be able to choose (you can in real life). But equally a random option is useful when players don't care.

Q) Should we implement a timer/turn limit for the players on the abridged version? - A timer specifically so people cannot get into the lead then waste time.
A) A timer would be in the spirit of the original game. "Filibustering" is for cads … [NB: look these words up in a dictionary …]

Q) Can we automate the bank?
A) As long as the bank does the job intended for it.

Q) Can players sell back properties to the bank or only mortgage them?
A) Good point. A player is free to sell a property to the bank for the price paid originally for it. A property may only be sold when it is free of houses and hotels, when it is the player's turn. When a property is sold, no improvements are possible in that colour group. A player must own all of the properties in a colour group before any improvements can take place.

Quentin Raffles

Further info from Kingsley Sage

Q) In the example on Study Direct, the functional requirements were organised into very high level APIs that grouped the functions of the software into "Game Requirements", "World Requirements" etc, is it okay to use this type of API approach for the formal document with Property Tycoon? Our team have identified many functional requirements but can't find a better way to organise the document other than this grouping approach. I'm worried it implies an object oriented design even though the requirements aren't supposed to indicate how the software will be designed!
A) The grouping is just a means of helping to organise a large amount of information in a logical manner. A requirements document and its structure have no specific bearing on an OO design. The design documentation that follows the requirements process does. The same requirements doc can be used to create both OO and plain procedural solutions.

Q) If we translate the domain requirements later on in the project (e.g. a few weeks later during actual coding) into functional/non-functional requirements, do we need to change the requirements document to reflect this?
A) If the changes are significant, it would be a good idea. But you could also just add an addendum to the original document rather than rewriting the whole thing. As long as your keep track of your requirements somehow, you should be fine. A good numbering scheme helps a lot.

Kingsley Sage