Requirements Creep

/~gibson/Teaching/CSC7426/L5-RequirementsCreep.pdf
Requirements/Feature Creep

This problem is concerned with looking at the life cycle when requirements creep during the development process.

What sort of procedures would you put in place if you knew that this would happen in advance?

How would this impact the design of your system?

Designs that are maintainable, evolvable and re-usable are more likely to be safe against requirements creep.
The Robot Problem: A first requirements specification

In a 2-dimensional grid/plane (n*m) there are either walls or spaces.

We represent the walls as ‘# ’ in the diagram (with spaces coloured into different partitions)

In such a grid we can place robots who can move *horizontally and vertically* but cannot move on top of a wall.

You need to calculate the minimum number of robots that are needed in order to be able to visit all spaces in any given grid (of walls and spaces)

In the example above, there are 3 partitions and so we need 3 robots.
Design for the future

You are to write a program that solves the robot problem, together with the test code that demonstrates that it works correctly.

Your program design should be ‘ready’ for the addition of new requirements/features or changes to existing requirements.

Try to predict what the ‘client’ could reasonably ask you to add or change, and evaluate how your design copes with such evolution.

 Explicitly choose one such evolution and restructure your solution, if you feel it could be improved to facilitate this evolution.

I will give you some additional requirements additions/changes once you have submitted your first solution.