<u>Design in XP</u>

<u>Evolutionary design</u>

Evolutionary design is designing the next part of the system once the previous one has been implemented. It means that the design is an ongoing progress, done at intervals throughout development rather than in one big up-front design at the start of the project. Its use means that if requirements change, no time has been wasted designing parts of the system that are no longer necessary. The design evolves as the system gets more developed, though this approach when not used with XP practices often results in ad-hoc design decisions and a worse design that makes code more difficult to alter in the future.

In Martin Fowler's 'Is Design Dead?' article,

<u>Planned design</u>

Planned design means designing the product before any major work is carried out on it. It plans and breaks down the task into smaller, more manageable tasks. It acts as a predetermined path for the development team to follow and keeps their work relevant and helps prevent them from straying from the necessary work, as I have often done in assignments I didn't plan, especially when tackling topics I'd never before encountered. To me, this design process looks at the goal point and finds the best route there, connecting the dots as you go. However, in the real world the requirements and environment mean that implementing one thing may be more difficult than originally anticipated, and the dots (milestones, required implementations etc) can move about, resulting in a greater distance between points, and more work. Following the path laid out by this design, in this case, is not as quick or efficient as following evolutionary design, where the next, closest step towards the goal is calculated at the time the previous one has been implemented.