

# HMS Process model

In our opinion, “Modified Incremental Development” model is well suitable for the development of our project i.e., Hospital Management System (HMS). The model stated above is not to be confused with Modified Incremental Delivery. Modified Incremental Development is a modification of the classic Incremental Development and takes some of the best features from other models in order to meet the needs of our project.

For a HMS, it's very important to make sure that the system to be developed is exactly what the client (the Hospital) has in their mind. So, it requires customer's interaction and feedback during the development process and we should be able to adjust the system the way they want to. Also, it's very important to make the system maintainable and flexible so that it would cope with any changes that might come in the Hospital management. So, waterfall model is clearly off the table and we need a slightly-agile approach. Best way to do this is to follow Incremental model. In an incremental model, we develop the system in increments and evaluate each increment before proceeding to the development of the next increment. Thus, we're dividing the whole project into parts, making it easier for developers to develop, managers to manage and also customers to comment and experiment on each part and have a clear understanding. Doing it this way is very client-friendly (because they'll be aware of how we're developing the project and also the progress we made) and also helps us in developing the system that was imagined by client because of the interactions with client after each increment, they could re-evaluate or even change their requirements for the next increment so that we could implement them.

Now the one small issue with the classic incremental model is that all (at least most of) the requirements aren't specified at the beginning. In the classic model, requirements of each increment are only given at the beginning of that particular increment so this doesn't give a whole picture of what the developer team is supposed to make at the end from the beginning itself. But in our modified model, we perform requirement specifications (and maybe the architecture design) in a standard waterfall approach, i.e., most of the requirements are specified upfront. This benefits the developers because they now have a clear picture of the final product they're supposed to make also including any further changes in the future.

But so far, our model is almost similar to “Modified Increment Delivery”; so, what's the difference? Well, the main difference is that in our model, we don't deliver the system to the customer after every iteration and it's good not to because, if we deliver only parts of the system, it's not usable because an HMS is a very connected system. For example, if in the 1<sup>st</sup> increment, we build the system enabling the receptionist to register an account for a new patient and to schedule doctor's appointments, and the patient to view their medical information, etc (basically meeting all the receptionists' and patients' requirement); it's not usable by the hospital because even though they can create an account and schedule appointments with the doctor; the reports, prescriptions and all the necessary information that should be stored and to be shown to the patient isn't there yet (because requirements of medical assistant, doctor and diagnostician aren't fulfilled yet). So, the patient will wonder what are they creating the account for if they can't view anything about their present visit. Similarly for all the other roles in the hospital (like for doctors, medical assistants, diagnosticians etc). Hence, the system can be put in use only after at least most of the important functionalities were implemented and those obviously can't be made in just 1 increment. However, despite of not delivering the system, after each increment, customer/client will be shown the system

we made so far and we let them comment and experiment on it to make any changes necessary. This way, it's good for both client and for the development team.

Hence, this is why, we think, "Modified Incremental Development" model is well suitable for the development of our project. Following are few advantages for choosing this model:

1. Requirements are mostly known upfront: Waterfall advantages (benefit over the classic Incremental development).
2. Rework in development iterations will diminish.
3. The cost of accommodating changing customer requirements is reduced because we're coping with changes during the development itself, so there won't be much to change or re-do after delivering the system (which would be costly).
4. It's easier to get customer feedback on the development work that has been done.
5. Because of this incremental approach, there'll be minimum chances of error and reduced redesign work (lower risk of overall project failure) because if one task fails, it'll be easy to identify and rectify the problem since it's just from one particular increment.