Thoughts about Systems Development

Lars Wentzel 2022-01-26

This is mostly about systems in large companies and organizations where complexity is a big issue.

Systems for large organizations seems to be a complicated thing. Usually, you start with a mix of bought systems, modified standard systems and systems developed specifically for you as inhouse development or developed by suppliers. They may be integrated in very different ways. Some of them are 20-30-40 years old and some are quite new.

“We need to replace all our legacy systems in this area with modern technology”
“For all our logistics system we shall use SAP” (or any other abbreviation)
“Our strategy is to go for MS” (Is that a decease?)
“This system is based on outdated software and must be replaced”
“Our ordering systems must be replaced”
“All systems should be built in Cobol and DB2”
“All system within the marketing area should be in AS400 and Synon”

Have you also seen all these dreams about a new clean start where everything is based on new technology that is also highly functional and productive? They never come true because they are out of the blue. Why?

- The complexity and functionality of existing systems are totally under-estimated
- New tools are not very productive
- New development takes time
- New development is costly
- You don’t know how to work with the new technology
- New developers have limited knowledge of the information, way of working, and dependencies.

Two industrial companies have each spent 15 calendar years each to replace their ordering systems. One bank has spent decades to replace a reporting system. Yet another company has tried to replace the purchasing system twice with fifteen years in between, no success. Public insurance wanted to replace their solution with SAP, it failed (public knowledge). Manufacturing systems to be replaced by new developed ones with hundreds of MSEK spent with minor result i.e., few new systems developed.

Social Services in Stockholm started a project to replace their old inhouse system Umbralla. The new system should be based on a BPM platform (Pega????). After two years and 250MSEK the new project was stopped and now the old system will be upgraded.

This means that you will have to do things stepwise. And then you easily get mixed up in new integrations and stepwise decommissioning of existing functionality. And remember, new systems quickly become legacy.

What do you do with an existing system when the new is being developed? To “freeze” functionality of the existing system could be seen as an option. But if new development takes years, you will be stuck with solutions that do not cover the changing needs. This means that you must allow changes in the “legacy” system. Then you are replacing something being changed.

It used to be said that the 80% of the cost of a living system comes after the initial development. So, if you invest 200MSEK now the total cost will be 1000MSEK.

So, remember that you will have to live with a mix of systems with different age and with different technology for ever.

Another problem you end up in when focusing on the technical environment: You just copy the old solution when you really need something new.

“Standard” solutions?
Popular is always, of course, to try the short-cut “bought standard solution”. But does it really solve your problem, or do you need to modify it? Or is it not solving the problem? You will anyway need a lot of new integrations as well as time and money to succeed.

SJ (Swedish Railways) recently introduced a new system for planning. It seems the ideas of this started 28 years ago. During the time different “standard” systems have been tried including the current one. Seeing it from the media information and from some blogs there seems to be several problems e.g., chaos in manning the trains and difficulties to use the system.

The Swedish regions Skåne and Västra Götaland will try to replace current medical administration with one gigantic system based on Cerner’s Millenium platform. The system shall replace 40 systems currently used. Cerner is said to have 29,000 employees. Recently Oracle with 132,000 employees has bought Cerner. So, the bigger the supplier, the better? The size of the project (Västra Götaland) started at 2 billion SEK and after adjustments the cost is now 3 BSEK. “The increase of cost is the underestimated cost of developing, decommissioning, and archiving of existing systems and delay of the implementation”. Have you seen this before? Not in this scale, perhaps, but it looks very much to me as an unfolding disaster. How can anyone specify the requirements on this system, initially and for the coming 20 years?

Success?
Is there any time a possibility to succeed?
This question does not have a simple answer.
But two examples:

1. A very large system for hospitals.
   There is a very large system for medical care information. I have no inside information, but I have followed the system for many years now. It all started in a very small scale almost 30 years ago handling test information. Currently it has 70,000 users and handles 4,3 million medical records and cover major areas of medical care information. Step by step it has been developed over all these 30 years and with a very small group of developers.

   I can imagine that managers would like to replace this for some reason (technology, ideology, strategy). But you may imagine, this is not easily done.

   A medical record folder being pulled from the records

2. Product data for vehicles.
   I have myself been involved in a system for vehicle product data (VPD). It all started with a prototype and then “a temporary solution” waiting for another system to be developed. This new system never appeared and VPD grew step by step over the years. Now it supports several tens of other systems with advanced services and has become the mayor supplier of this

information. Success factors: Special technology, a small and dedicated staff, knowledge, and ability to solve new upcoming problems.

Then, of course, IT management want to replace it for ideological reasons.

Decisions
How are decisions taken about these costly investments in software? Well, I really don't know. This is usually not an open and transparent process where different views and opinions can meet. It looks more as if ideology (strategy), vendor influence, external "advisors" decide what is chosen. And the basic hard homework is not done i.e., to find all the important functionality, to perform the proper Proof of Concept, to find out how to integrate with the current systems, and to roughly estimate the time and cost. And there seems to be a lot of ignorance about what information processing and systems are all about from top managers and not seldom from IT managers as well. Finally, management on different levels are replaced all the time.

This summer I listened to a Swedish top manager at Pfizer who said that they had Kahneman training of the top managers so that they could avoid wrong decisions. I don't know if that is really needed. Maybe, it is enough with a good checklist, knowledge, transparency where different opinions can meet, hard work and ability to see more than one alterative.

Research required
All these mistakes in systems development are costly. For the company (organization) and for society. And furthermore, the ability to change and to implement needed functionality is hampered. Then the increasing lack of developers makes this even more problematic.

I would very much see real research done in this area. How should decisions be taken? What should you think of when selecting a new solution? Is there a good idea to migrate an existing solution?

There is a US company, Standish Group, that has been working on this for a long time and they have a big database with projects. They have some valuable advice to give e.g., that it is important with a good and active sponsor, and that you should run only small projects. Still, I find it a bit limited not covering all the problems.

Who will pick up this idea of starting research? Perhaps start with finding what has been written about this.

What are the key issues?
This is really the question to answer, and I can only list some of my thoughts.

- Understanding the problem.
- Understanding the information.
- Productivity. This is really an issue. There is a big lack of developers. Routine development must become much more efficient. And what about complex logic? There has been a lot of focus on methodology. This usually only adds bureaucracy e.g., Scrum administrators.
- A qualified team that will be there for a long time.
- More than one alternative. Always consider upgrading the old system as an alternative.
- Do the hard work to find what should be done. Especially key functionality. I worked for a company where the architect drew his diagram, estimated a cost for X SEK and the shortly afterwards left the company. His estimate was the base of the offer to the customer to a fixed price. The outcome was a cost four time as high as the estimate and not a very stable system.

- POC (Proof of Concept) must be properly done covering the difficult problems.
- Service orientation. Try to communicate through on-line services.
- Modules. Can you split up the problems?
- A clear target.
- A dedicated sponsor.
- Measurements. Cost, user efficiency.
- Follow up to learn. What did you accomplish? Don’t forget your mistakes.