

# The new informatics revolution - not

Per Flensburg  
University West  
*per.flensburg@hv.se*

**Abstract.** Systems development seems to be taught in a very traditional way in the Swedish universities. It is supposed that an inhouse development starting from scratch will be at hand. This is shown using an investigation of current books in the area, of contemporary educations in systems deveopment. But the needs from the business word are different and this is shown in an investigation of job advertisments in Sweden. The conclusion is that informatics as subject is at a deep crise: We educate students for a work that was at hand in the 80's and not for the 21<sup>st</sup> century!

## Background

The subject of informatics was in Scandinavia coined by Börje Langefors in the middle of the 60's. Börje realised that "data" and "information" was not the same. Information was used in a company, it is not enough to have access to it, you have to know how to use it also. Börje argued that "information" was about the same thing as "knowledge". The enterprise need for information was central for him and he developed a theory for its capture, processing and use (Langefors B 1966). This area was later called informatics.<sup>1</sup>

At the same time there was an area within business economy called "administrative rationalisation" aiming at reducing the cost for control and management of the enterprise. The area started with F W Taylor and build upon the principle of separating planning of work and doing work. (Taylor 1911) Computers are very good at processing specified instructions; in fact that's

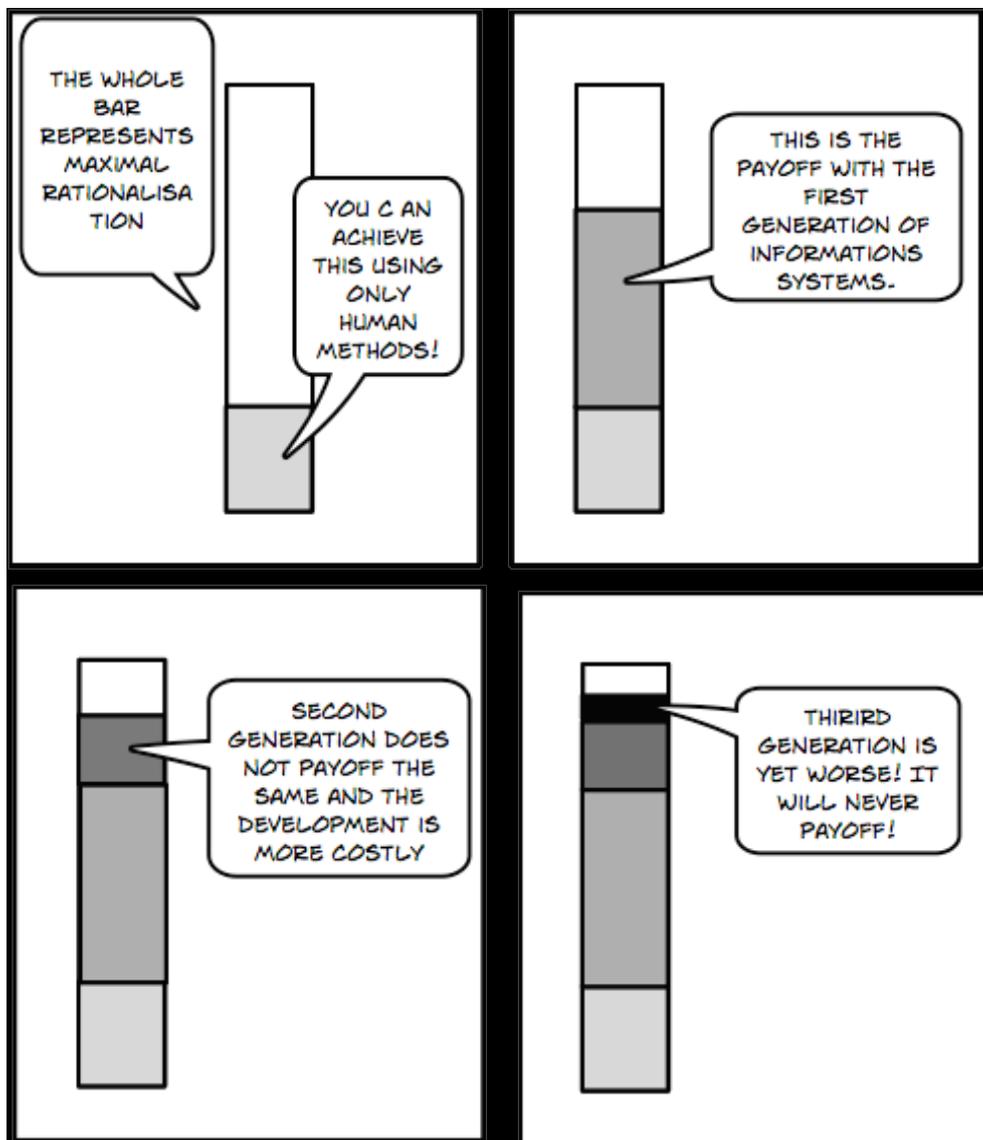
---

<sup>1</sup> In fact Kjell Samuelsson first introduced the concept of informatics, but it was used in the context of information science rather than administrative data processing

exactly what a program does. Hence it was natural to introduce the use of computers within this area.

At that time computers were very expensive and due to economical reasons a physical allocation with numerical analysis, which also used computers, were done. In the 70:s computer science, focusing on efficient use of computers developed. Informatics had its focus on use of information in enterprises, but both of them used computers. It was inevitable that conflicts raised.

Langefors assigned the term “systems development” to the process where a good and efficient information system was created for the company. In (Langefors B 1966) he presented the ideas that was to lead the area for at least 40 years! It was from the beginning a central area in informatics and thousands of students have been tormented with different versions of systems development models, systems descriptions techniques, stages, phases and waterfalls! But in connection with the hype about the millenium shift it struck me suddenly: Systems development is obsolete! It will never more occur in the way we are teaching it! My arguments is shown in the pictures below.



The first time a computerised information processing system is introduced in the organisation it pays off very fast. This is due to the fact that the most common and easy transactions are dealt with (the 80-20 rule). The cost reduction for administration is tremendous. But in the next generation, we face some more difficult problems. Part of the transactions not dealt with earlier must now be taken care of. They are more complicated and hence the system development takes longer time and costs more money. Also the available cost savings are considerably less than before. We have higher cost and lower return on investment, hence the pay-off time is considerably longer. Taking about the third generation, this trend is more obvious and probably these systems will never pay off! However, it is to be noted, that the indicated height of the bars in the figure have no correspondence in reality. This quantitative investigation, to my knowledge, has not been undertaken. However, I am convinced, that after some generations, the new systems will never pay off.

With this model in mind I have during the years been more and more astonished over the fact that we still continue teaching the students as if we still were in the life cycle model and the first generation system<sup>2</sup>! So the question I address in this paper will be: "Are teaching in systems development in the informatics area still based on the assumption of inhouse development of the first generation system? And is this what the employers want?" In order to get an indication of the answer to this question the following investigations are carried out:

1. Inspection of books in the systems development area
2. Inspection of the content in the teaching in systems development at Swedish universities.
3. Inspection of employment advertisements in Sweden

The first investigation is rather easy to do. Almost every book in the area is published at Studentlitteratur. Therefore I think a good starter could be looking at those books classified as systems development. In most cases it is enough to look at the table of content, but of course personal knowledge of the authors and my own reading of the book have also an impact.

The result is shown in the table below. All books were in Swedish, since systems development is usually taught in the beginning of the education. I have selected all books found during the header "systems development" at the web-site of the company. There were 26. Of them 6 were omitted because they dealt exclusively with programming. The remaining 20 belongs most of them to the informatics area, but a few can be classified as software engineering. The reason to believe these books in some sense represents the area of systems development

---

<sup>2</sup> However, it is to be noted that the model we teach today are considerably more sophisticated than the simple ones used in the 70:s

is simply the fact they are still in stock and hence used in the education. Some of them are rather old but since Stuentlitteratur is based on ordinary economic considerations, I think it is a reasonable assumption that they mirror the Swedish teaching in systems development.

<b>Title</b>	<b>1st gen</b>	<b>Lifecycle</b>
Eiderbäck 2007	Yes	No
Nyström, Fernström et al. 2006	Yes	Yes
Magnusson and Olsson 2005	Yes	No
Brandt 2004	No	Yes
Löwgren and Stolterman 2004	Yes	No
Lindegren 2003	Yes	No
Wiktorin 2003	Yes	No
Lunell 2003	Yes	Yes
Fagerström 1999	Yes	Yes
Gulliksen and Göransson 2002	Yes	No
Ottersten, Berndtsson et al. 2002	Yes	No
Apelkrans and Åbom 2001	Yes	Yes
Mathiassen et al 2001	Yes	
Ortman and Flink 1999	No	No
Axelsson and Goldkuhl 1998	Yes	
Brandt, Carlsson et al. 1998	No	No
Hägerfors and Bryngelson-Ohlsson 1995	Yes	No
Andersen, Helleskog et al. 1994	Yes	Yes
Axelsson and Hedefjäll 1993	Yes	
Goldkuhl and Röstlinger 1988	Yes	

We see that all of them except three, were based upon the assumption of creating a new in-house system. One exception was about project management, apparently in a general way, since systems development or any IT-related terms was not to be found in the table of content. The other two were about systems maintenance and standard systems. The second: “To choose and maintain

standard systems” seems to be based on the assumption that the company buys a standard system from scratch.

I also tried to figure out if the book was a patron of the systems life cycle model and it turned out that 6 was and 9 was not. So finally we might get rid of that model.

## **Teaching investigation**

The examination of bookd just give a hint that further investigations might be worthful. In the next step we investigate the real teaching in the Swedish universities. This investigation is presented in Nåfors & Johnson (2008). First they looked at every education program dealing with “systems development”. 66 different programs were found. They had examination within 14 different areas, where informatics and computer science each had 23 and 20 programs. The next area, data- and systems science had 4 different programs. After a categorisation, based upon Grounded Theory, 30 programs within informatics were identified. In these programs 252 courses were found and among them 56 courses dealing with systems development. This was found by looking at the course plans. However, there were in total 105 courses where the course plan was not available, so these courses were omitted. Among them there were several that seem to be about systems development, due to the name. The course plans revealed the following concerning learning outcomes:

Within the category “The students should be able to critically judge”:

- Users role in the systems development process (3 goals)
- Integration in the systems development process (2 goals)
- Choice of different models for systems development (2 goals)

Within the category “The students should be able to”:

- Use knowledge of the role of information and information systems within organisations (60 goals)
- Apply and describe different modelling techniques (25 goals)
- Apply skills learned in a project (20 goals)
- Apply object oriented systems development and UML as modelling language (16 goals)
- Define and use central concepts within the subject area (12 goals)
- Describe and choose architecture strategies and their use (11 goals)
- Apply design modelling within systems development 11 goals)
- Describe usability factors, user centred systems design and human computer interaction aspects (10 goals)

- Apply and understand the usefulness of using requirements specification in the systems development process (9 goals)
- Apply and use different evaluation models (6 goals)
- Apply and use web services and service oriented architecture (6 goals)

Within the category of “The students should have knowledge about and understanding of”:

- The role of information and information systems role in the organisation (9 goals)
- Different models in systems development (6 goals)
- A process oriented working process (6 goals)
- The role of the systems architecture (5 goals)
- The use of UML and object orientation (3 goals)

The conclusion is interesting. Understanding the role of information and information systems role in the organisation is by far the most important learning outcome (180 out of 240). Looking at learning outcomes concerning classical systems development we find 49 learning outcomes, while there are only 11 outcomes dealing with modern service oriented development. However, many of the courses where no course plans were found, indicate that they are oriented towards modern service oriented architecture. Hence, I conclude that the teaching in systems development seems to be in the beginning of a transition phase from inhouse first time development to more service oriented architecture.

## **Labour market investigation**

Seen in this light, the demand on the labour market for systems developers should not be very high, in fact it ought to be almost zero. Therefore I did a small investigation and to my big surprise, the result was on the contrary: A very high demand!

The procedure was as follows:

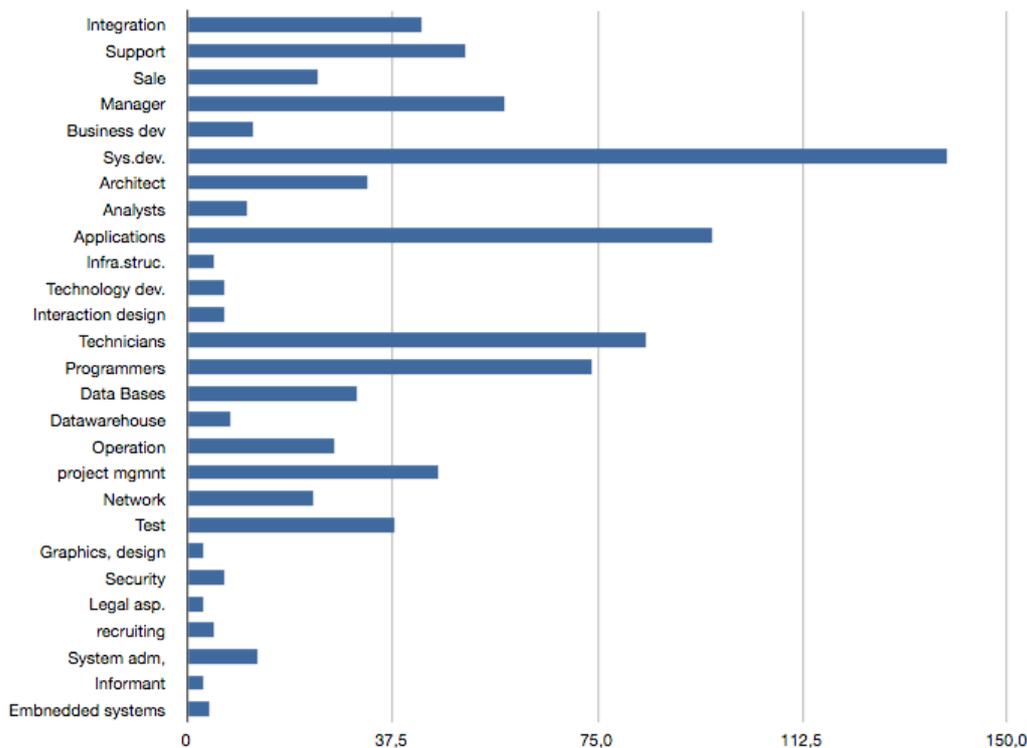
First I got a list of names of possible job titles and descriptions from the Swedish Public Employment Service (AMS). A system developer was found to have to following occupations:

- GIS-developer
- IT-architect
- Project manager-IT
- System architectt
- System designer
- System developer
- System engineer
- System constructor
- System investigator
- Webb developer
- Business developer

It is also to be noted that AMS writes (my translation):

The data- and IT-area is dynamic and changeable. We see a growing integration of computer platforms with phone- and communication systems. In certain area the change is so big that knowledge become obsolete in less that a year. The borders between the occupations are floating and the job title used can change between work places.

Having this in mind, I went to the biggest job-site in Sweden where system developers was to be found, namely at Computer Sweden, the biggest newspaper within the IT-area having a special focus on IT in private companies. The job-site was visited on the Feb 8th 2008 and 594 jobs were found. According to the headline each of them was classified in one or more categories, dynamically created. Base for the classification was the title in the database. About ten of the entries have titles not saying anything about the position, so they were omitted. The classification was done three times in order to compare every advertisement with the full set of categories. The result can be seen in the diagram below.

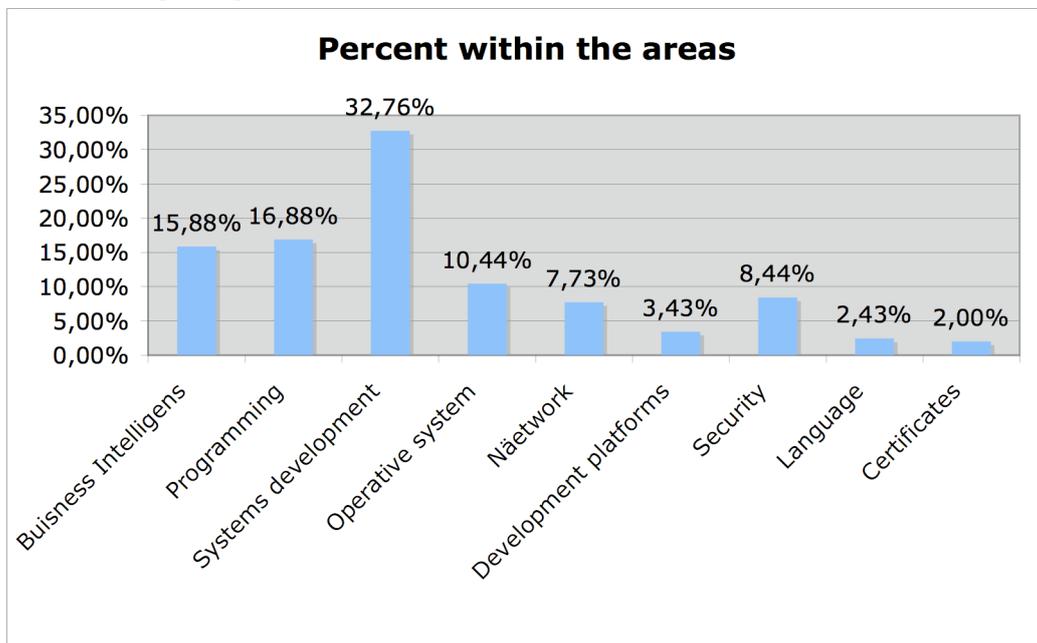


It turned out that the classification was by no means simple. Very often an advertisement fitted into several big categories. For instance both “system developer”, “Java-developer” and “System developer with C/C++” was classified as “systems developer”. The two latter was also classified as programmers. It was

also hard to delimit between applications and programming/development platforms. In general I have favoured applications before development platforms. Biz-Talk, SQL, Windows and Unix were considered as applications while PHP, Java and XML were classified as programming languages.

The top categories are system developers, applications, technicians, programmers, managers, project management and integration. A similar investigation was done a year ago and it shows about the same distribution, except for a considerable increase in integration.

This investigation seems not to be consistent with my idea of systems development being an obsolete occupation. However, examine some of the advertisements, indicates that “system developer” has a very different meaning in these texts. In most cases “systems development” means “systems maintenance” according to the taxonomy we usually use. A further investigation into the area of the meaning of the words in position advertisements revealed (Kjellström, 2008) the following categories.:



They were deduced from a closer study of the text in a sample of the advertisements, describing the area of the position announced. However, a further analysis of the meaning behind “systems development” was not possible to conduct, since the text gave no clues about the meaning. For instance a position “systems development with .Net” can mean very different things.

We might also get an explanation to the fact that in the advertisements programming is always asked for, but in face to face talk, other types of knowledge, such as knowledge about the business and business processes are asked for. In the advertisements the positions are for beginners, but in F2F-talk we talk about the more advanced positions. Traditionally, you always start by

maintaining the old systems and after that you can advance to higher positions and greater responsibility.

## Conclusions

This little investigation indicates that informatics is at crises. Our education in one of the most central areas is old-fashioned and based upon circumstances no longer at hand. For many years we have not realised this, but today when the student applications has decreased to just a fraction of those in 1999, we might finally realise that something has to be done. In fact, we have detected signs of a re-orientation towards a service oriented perspective.

The challenge is to rethink informatics from the very beginning. We must start it over again, facing a new information society, driven by IT, but in fundamentally different ways than we are used to. I feel privileged for the opportunity to take part in this!

## References

- Andersen, E. S., A. Helleskog, et al. (1994). Systemutveckling : principer, metoder och tekniker. Lund, Studentlitteratur.
- Apelkrans, M. and C. Åbom (2001). OOS/UML : en objektorienterad systemutvecklingsmodell för processororienterad affärsutveckling. Lund, Studentlitteratur.
- Axelsson, K. and G. Goldkuhl (1998). Strukturering av informationssystem : arkitekturstrategier i teori och praktik. Lund, Studentlitteratur.
- Axelsson, L. and M. Hedefjäll (1993). Praktisk datamodellering : ta greppet om begreppen. Lund, Studentlitteratur.
- Brandt, P. (2004). Guide till systemförvaltning. Lund, Studentlitteratur.
- Brandt, P., R. Carlsson, et al. (1998). Välja och förvalta standardsystem. Lund, Studentlitteratur
- Eiderbäck, B. (2007). Extremprogrammering. Lund, Studentlitteratur.
- Fagerström, J. (1999). Objektorienterad analys och design : en andra generationens metod. Lund, Studentlitteratur.
- Goldkuhl, G. and A. Röstlinger (1988). Förändringsanalys : arbetsmetodik och förhållningssätt för goda förändringsbeslut. Lund, Studentlitteratur.
- Gulliksen, J. and B. Göransson (2002). Användarcentrerad systemdesign : en process med fokus på användare och användbarhet. Lund, Studentlitteratur.
- Hägerfors, A. and A. Bryngelson-Ohlsson (1995). Att samlära i systemdesign. Lund, Studentlitteratur.

- Langefors B (1966). Theoretical Analysis of Information systems, I & II. Lund, Studentlitteratur.
- Lindgren, H. (2003). Programvaruprojekt : stabilitet, användbarhet och inkrementell utveckling. Lund, Studentlitteratur.
- Lunell, H. (2003). Fyra rundor med RUP : förberedelse, etablering, konstruktion, överlämning. Lund, Studentlitteratur.
- Löwgren, J. and E. Stolterman (2004). Design av informationsteknik : materialet utan egenskaper. Lund, Studentlitteratur.
- Kjellström J: En kartläggning av arbetsmarknadens behov av IT-utbildningar i Sverige, University West (2008)
- Magnusson, J. and B. Olsson (2005). Affärssystem. Lund, Studentlitteratur.
- Mathiassen, L, Munk-Madsen A, Nielsen P A, Stage J: (2001). Objektorienterad analys och design. Lund, Studentlitteratur.
- Nyström, A., B. Fernström, et al. (2006). IT-byrån : handbok om hur affär, IT och förändring hänger ihop. Lund, Studentlitteratur.
- Nåfors A, Johnson M (2008): En kartläggning av Systemutvecklingsutbildningar i Sverige med tyngdpunkt på Informatik – Vad är det man utbildar inom?, University West (2008).
- Ortman, L. and M. Flink (1999). Praktisk ledning av förbättringsprocesser : inriktad på samverkan mellan resultat, process och team. Lund, Studentlitteratur.
- Ottersten, I., J. Berndtsson, et al. (2002). Användbarhet i praktiken : praktiska handgrepp, grundbegrepp och tankemodeller. Lund, Studentlitteratur.
- Taylor, F. W. (1911). The principles of scientific management. New York, London,, Harper & Brothers.
- Wiktorin, L. (2003). Systemutveckling på 2000-talet. Lund, Studentlitteratur.