

SYSTEMS MODEL AND LIFE

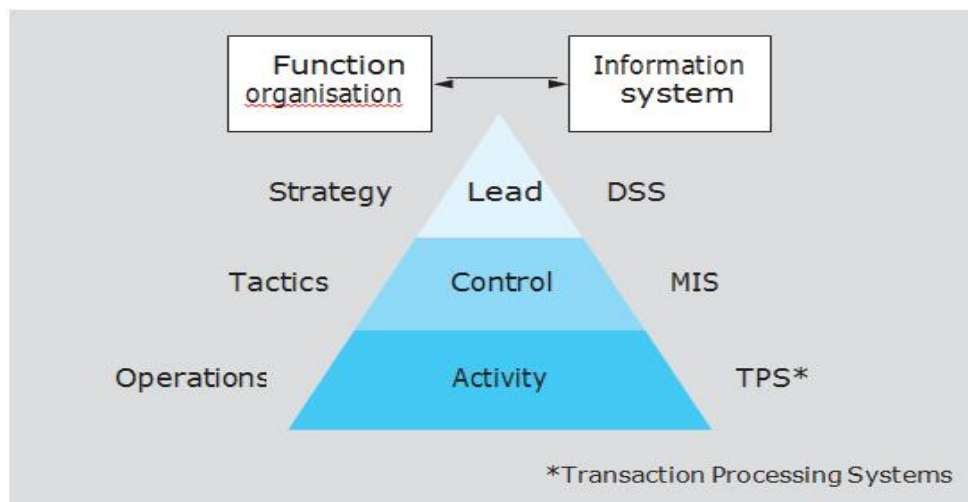


Figure 1. Anthony's pyramid

Source: *Information and Knowledge Management* by: Elsevier Butterworth-Heinemann Linacre House, Jordan Hill, Oxford OX2 8DP 30 Corporate Drive, Burlington, MA 01803 First published 2005 ISBN 0 7506 6688 9

CYCLE

From the late 1960s it was realized that systems development actually consisted of well-defined stages, and a 'life cycle' view of systems emerged that formed the basis of many different methodologies for systems development (Galliers et al. 1999).

System development should be viewed as continuing process in order to:

- review and correct earlier errors and misconceptions
- revisit and retune the original specification in the light of changing requirements
- deal adequately with the problem of a growing number of systems involving increasing amounts of maintenance.

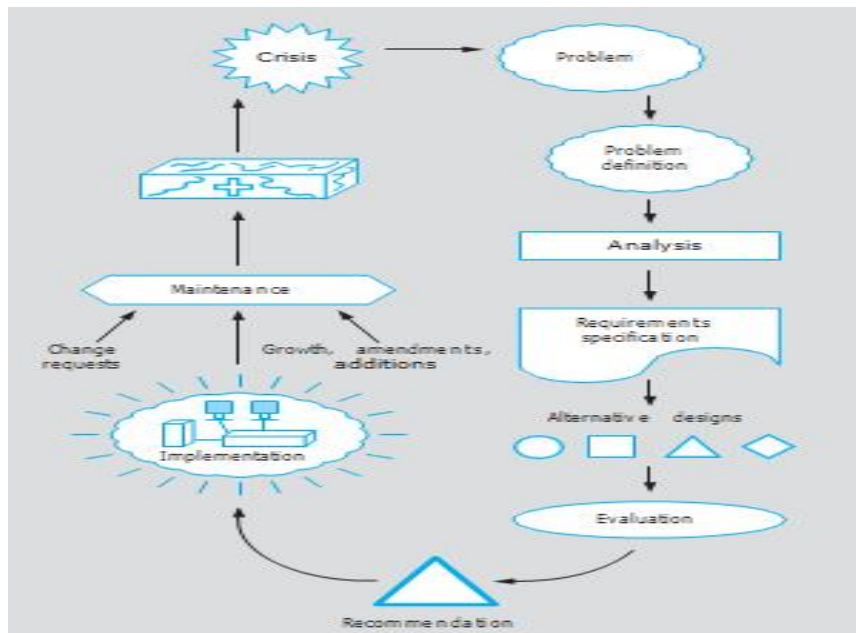


Figure 2. The system life cycle

Source: Information and Knowledge Management by: Elsevier Butterworth–Heinemann Linacre House, Jordan Hill, Oxford OX2 8DP 30 Corporate Drive, Burlington, MA 01803 First published 2005 ISBN 0 7506 6688 9

ROLE IN DEVELOPING INFORMATION SYSTEMS

If you are not an IT manager, where do you fit into this process? You may not realize how important you are. Recent years have seen the growth of user involvement at every stage of systems development.

As the impact of systems development has become visible and organization wide, organizations have slowly realized that it has a human dimension, and that ignoring this can wipe out all the advantages of your expensive new system.

Many organizations are now battling with the problem of so-called legacy systems. This is a system that was developed 20-odd years ago which was designed to solve the problems the organization faced at that time.

The trouble is that 20 years on, they are out of step with evolving business needs and can hold back organizations that want to apply a more up-to-date set of routines. They are now seen to be inflexible,

expensive to maintain and even more expensive to replace.

This has all lent urgency to the need for genuine user involvement in systems design, and few would dispute the necessity for this. But mistakes still happen.

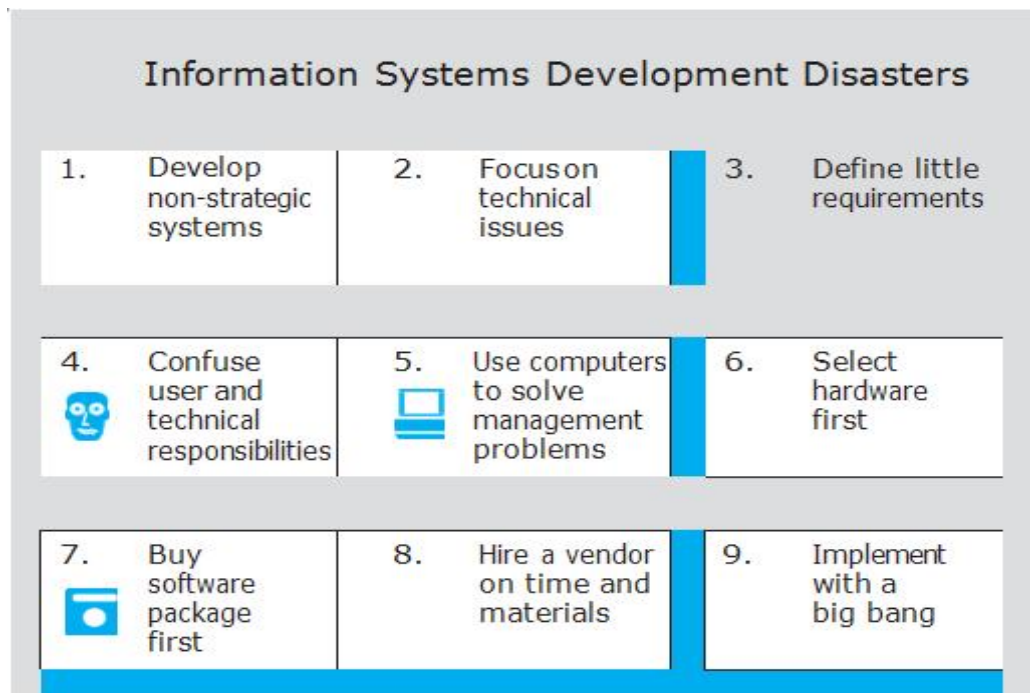


Figure 3. An information systems disaster menu

Source: Information and Knowledge Management by: Elsevier Butterworth–Heinemann
Linacre House, Jordan Hill, Oxford OX2 8DP 30 Corporate Drive, Burlington, MA 01803
First published 2005 ISBN 0 7506 6688 9

Lytle (1991) devised an information systems development disaster menu, shown in Figure 3, that still holds good. As you can see, it shows all the things you shouldn't do when developing computer systems. If you do, then you're heading for disaster.

Let's turn the negatives from Figure 3 around and see what happens.

1. **Develop strategic systems.** What are the key *strategic information areas* for your business? These are your key business critical systems.

2. **Don't focus on technical issues.** Systems are not a matter of hardware and software; they are a matter of the *right* hardware and the *right* software, selected on the basis of user needs and organizational critical success factors..

3. **Take time to define requirements at 'big picture' and operational levels.** This is a key area of a manager's job. Welcome it as a real opportunity to examine quite critically what you are doing now. Do you still need to do it at all? Are there other, better ways of achieving your objectives? This will involve you in the following activities:

- Analyzing all the business processes that you manage. For example, if you're an HR manager, this will include such areas as recruitment, appraisal and reward, workforce planning and job analysis and design. How do the functions that you manage fit into the wider organizational picture?
- Documenting the type of information, you need to carry out your various tasks. Are you receiving everything you need, in the most efficient and timely way? Are you getting too much information, information you don't need at all or need less often?
- Drawing a picture of the data that flows in and out of your unit (don't forget to include informal data sources). What, and from where, are your data feeds? Who else receives the same information? What do you do with the information you receive? If you process it in some way, how is this done and what are the outputs? Who are they delivered to?

4. **Be clear about user and technical responsibilities.** Make sure that responsibilities are clearly defined at the outset, with staff allocated the roles that they are best qualified to do. That way you can build up co-operation and mutual respect, not mutual antagonism.

5. **Use management, not computers, to solve management problems.**

Too often, problems that are actually related to poor management are conveniently blamed on 'the system'. Computers can do lots of things to improve your data management and information flow, but they can't resolve problems of organizational culture or personality clashes. Do what you can to get these issues resolved before your user specification gets underway.

6, 7 and 8 **Select hardware and software to fit the requirement and be specific about any customization required.** Adopt the motto 'focus on functionality'. Vendors are experts at showing off their systems to their best advantage, but will the system do what you want it to do? Can the software be customized, and will the vendor do this? Will they need to involve third-party suppliers?

If customization is required, get this specified in terms of activities and costs. Hiring on a time and materials basis is a recipe for a long drawn out, expensive and increasingly sour relationship.

9 **Beware 'big bangs'.** It is rare now for a complete system to be developed in full before live operation, and for good reason. It takes time to develop a complete system, and while this is happening there will inevitably be evolution and change in user requirements.