Looking over the Edge

Requirements Engineering in Private Banking

Requirements engineering plays a crucial role in the success of a project, whatever the next technology and regardless of whether the bank's biggest concern is compliance, digitization, or outsourcing. Requirements engineering commonly refers to the process of identifying and analyzing, documenting, verifying, and maintaining software requirements.

To illustrate the crucial role of requirements engineering, recall the story of the French aircraft carrier in the 1990s. Its flight deck was too short for the aircrafts that France had bought for the vessel. Did "launching aircrafts" not make it onto the requirements list?

Requirements engineering for wealth management and private banking should distinguish two types of requirements. The first type are the requirements of the system itself: What is the system required to do, and how should it behave? The second type are the external requirements: What is required of the system's environment, and what restrictions or conditions apply to the environment?

For the first type of requirements, the IT engineering discipline offers a variety of tools and methods, including the prototyping and visualization techniques mentioned in the article above. To come up with correct specifications, the project sponsor is well advised to engage end users and domain matter experts—people who understand what the system is supposed to do and who can reliably verify the correctness and completeness of the specifications once they have been produced. The following example illustrates what can happen if the requirements engineering process is flawed. A few years ago, the Spanish Navy commissioned a new submarine. Now nearly finished, the submarine is too heavy and may not resurface if sent to sea. Was there a calculation error in the specifications that went unnoticed?

For the second type of requirements, the focus shifts to the environment. The objective is to analyze what is required of the environment to make the system work as intended. Many projects fail due to incorrect environmental assumptions and lack of contextual detail. A recent example is the case of the French national rail operator who ordered a new generation of regional trains, only to find out that they are too wide to fit many of the country's stations.

Don't Forget the System Environment

After an elaborate Request for Proposal and evaluation process, the wealth management division of a large bank purchased the performance measurement system of the preferred vendor. The system supported a wide range of performance measurement methods. The user interface was flexible enough to make choices regarding portfolio selection and time periods. The project team was convinced that the performance figures would be a real value-add for the client portfolio valuation reports.

However, once integrated into the IT landscape of the bank, the system did not behave as expected. The performance figures were so deficient that they could not be included in the reports to the clients. What had gone wrong? The project team investigated the problem and discovered that the data that were fed into the performance system were erroneous. Some positions did not match, some were valued incorrectly, and some were missing. The reason: The requirements specification did not include the systems that provided the data for the performance calculation.

Why do external requirements play an important role for projects in wealth management and private banking? In contrast to the banking models of the 1980s that centered on products, today’s models are client centric. This means that the majority of business processes cut across systems, data pools, and domains. As a consequence, the majority of the requirements in a project relate to the environment of a system rather than to the system itself. Consider, for example, an order management system: It depends on client data, instrument data, reference data, compliance computation (suitability), risk management (availability), tariffs and fees, trade execution, and transaction settlement.

Engineering external requirements is challenging. The engineers need to check that the system or process fits into the existing landscape and works with multiple channels. They need to understand and control myriad dependencies that may be triggered by a change to an existing system. They need to avoid creating new dependencies inadvertently. In addition, and equally important for the bank, the engineers should exploit synergies by identifying and combining similar requirements, thereby preventing the proliferation of isolated solutions. Furthermore, they should consider run-the-bank, usability, and maintenance requirements of the environment. Only experienced requirements engineers master these challenges successfully.

Conclusion

For many senior managers, requirements engineering is the most relevant part of a project. Numerous projects fail because of flaws in requirements engineering. And although failure due to technical problems may be viewed sympathetically, failure due to flawed requirements engineering harms the relationship between IT and business.

Wealth management and private banking projects should pay special attention to requirements that relate to a system’s environment. These requirements are the most challenging, but they are also the most rewarding for exploiting synergies and streamlining the system landscape.