Xeratec

2023

CONSTRUCTION COMPANIES JOURNEY OF DIGITAL TRANSFORMATION

Observations from those companies that went from $100 million to $500 million and beyond.

TABLE OF **CONTENT**

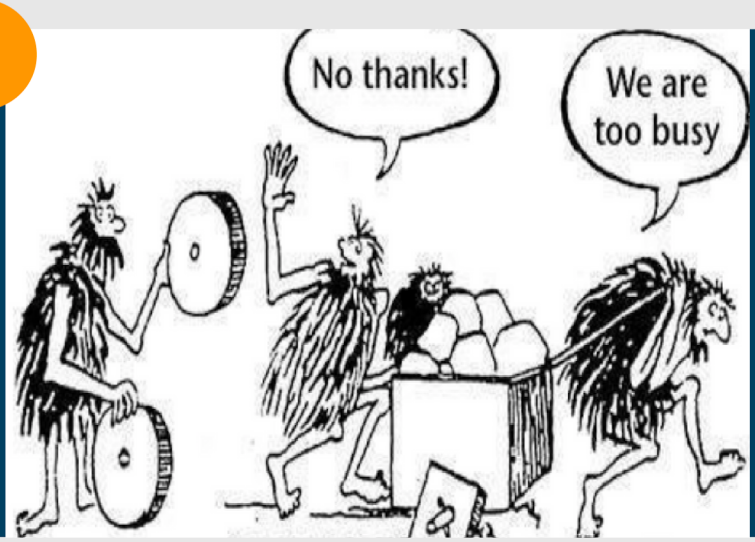
The Construction industry is a dynamic and ever-evolving field that demands precision, efficiency, and accurate resource management. As it continues to grow and becomes increasingly competitive, companies are realizing the need for advanced technologies.

GROWTH DOESN'T COME WITHOUT CHALLENGES

Construction companies exhibit diverse cultures, scopes, and operational methods, yet they often share a common origin story. The majority of these companies were founded or led by individuals who prioritized completing projects and getting personally involved in the work rather than establishing a scalable business.

The intriguing reality is that these companies experienced rapid growth over time. However, many of them continue to rely on makeshift solutions, or what are fondly refer to as "workarounds," which persist in numerous departments throughout the construction industry.

"Sometimes we are too busy putting out fires to realize the fires could have been prevented."



GROWING WITH INTENTIONALITY

Going from $100 million to $500 million and beyond doesn't happen without investing in the right technology solutions.

When leaders from various construction companies with a revenue of $250 million and above were interviewed, three out of four of them identified the adoption of technology as a key factor in their growth and ongoing success. While there are undoubtedly multiple variables contributing to a company's growth, the adoption of technology has emerged as a significant catalyst for many companies.

Custom software and related tools play a crucial role in fostering the growth of construction companies by streamlining processes, optimizing resource allocation, and enhancing collaboration. Tailored software enables construction firms to effectively manage projects, make well-informed decisions, and boost productivity. Additionally, these technologies facilitate seamless communication and collaboration, ultimately leading to improved customer satisfaction.

By harnessing the power of custom software, construction companies can maintain a competitive edge, deliver high-quality projects, and adapt to changes within the industry.

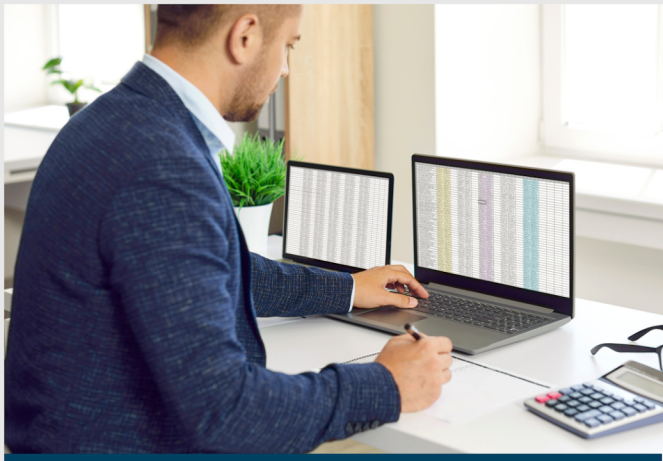
HELPING TECHNOLOGY LEADERS MAKE THE JUMP

"While there are numerous programs and processes that technology leaders need to address, two commonly observed programs where technology leaders have achieved significant success in moving beyond are Microsoft Excel and Access."
-**Tim Barrens, CTO**

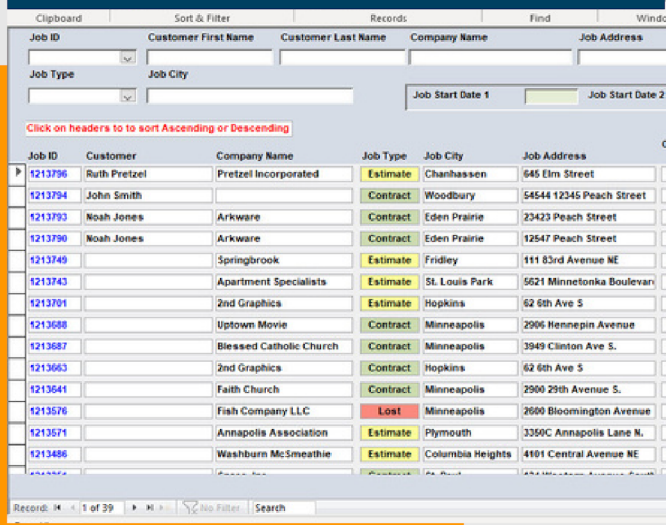
Transitioning from Excel and Access

Transitioning from Excel and Access-based processes to more advanced systems can be challenging but ultimately rewarding for organizations. While Excel and Access have been popular tools for managing data and automating processes, they do have limitations when it comes to:

1. Scalability
2. Data Integrity
3. Collaboration and Accessibility
4. Automation and Efficiency
5. Reporting and Analytics
6. Integration and Connectivity



70% of construction companies still use Excel or Access for their primary tool for project management.



WHAT DO THE STATS SAY

43%

A survey conducted in 2017 by *Intersect Associates* reported 43% of Microsoft Access users reported downtime which costs on average $10,000 per downtime.

35%

Another survey conducted by *Intersect Associates* reported 35% of Microsoft Access had data corruption problems. The average cost of the data corruption was $5,000.

22%

A *MarketWatch* article cites a study done by the professor of IT Management at the University of Hawaii, Ray Panko, that 22% of construction companies have been fined due to Excel errors.

90%

The study done by *Ray Panko states,* "Close to 90% of spreadsheet documents contain errors, a 2008 analysis of multiple studies suggests. Spreadsheets, even after careful development, contain errors in 1% or more of all formula cells."

33%

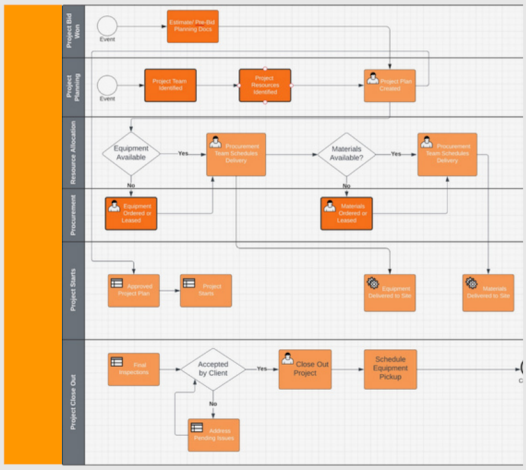
A survey conducted by *Varicon* found that 33% of construction companies lost clients due to Excel errors.

STEP-BY-STEP GUIDE TO SUCCESS

Embarking on a successful digital transformation requires an honest assessment of your current situation and a willingness to embrace change. Create a clear roadmap with specific objectives, secure executive support, and foster a culture of innovation. Collaborate with technology partners, iterate through testing and feedback, and continuously monitor progress to ensure a successful transition to a customized solution. Here are a few steps to consider as you take this digital transformation journey:

1. Stakeholders

Identify the stakeholders, both system and human, who will benefit or are impacted by the transition to custom software. List their responsibilities and how they might change with the implementation of a new technology.



2. Current Process

Engage with stakeholders to map out current process. Use a process mapping tool such as as lucid-chart or Visio to document.

3. Future Process

After evaluating current state and the recommended changes map out the future state. Ensure stakeholders confirm changes and are bought in; this will help with adoption and implementation.

4. Requirements

Once you have identified your ideal process, there are several requirements you will need to take into consideration. Each company has unique requirements making a custom application beneficial. Requirements such as:

4.1. User Requirements

4.2. Functional Requirements

4.3. User Experience Requirements

4.4. Architectural Requirements

4.5. Security Requirements

"Requirements are the bridge between the business and the technical teams. They ensure that the final product meets the needs of both the users and the business."
- Alistair Cockburn

A CLOSER LOOK AT REQUIREMENTS

4.1. User Requirements

- **Identify Stakeholders** (Personas)
- **Conduct user research:** Gather insights about user needs, preferences, and pain points through surveys, interviews, and observation.
- **Prioritize requirements**: Collaborate with stakeholders to rank requirements based on importance and feasibility.
- **Document the requirements**: Create a clear user requirements specification (URS) with detailed descriptions, use cases, and acceptance criteria.
- **Validate and verify requirements**: Review and revise the requirements document with stakeholders to ensure accuracy.
- **Maintain traceability**: Establish traceability between user requirements and development stages for consistency.

Here are some examples of user requirements:

- As a project manager I need to be able to edit my employees time.
- As a team member I need to be able to add time against activities that I work on throughout the day.
- As a project manager I need to be able to upload documents such as photos from the field to my daily report.

REQUIREMENTS CON'T

4.2. Functional Requirements

- **Identify the business needs and objectives**.What are the problems that the software is intended to solve? What are the goals that the software is supposed to achieve?
- **Break down the business requirements into small, more specific chunks**. This will help you to identify the specific actions, processes, and features that the software needs to have.
- **Write down the functional requirements in a clear and concise manner.** The functional requirements should be specific, measurable, achievable, relevant, and time-bound.
- **Prioritize the functional requirements.** Not all functional requirements are created equal. Some are more important than others. Prioritize the functional requirements so that the most important ones are implemented first.
- **Get feedback from stakeholders.** Once you have written down the functional requirements, get feedback from stakeholders to make sure that they are clear, accurate, and complete.

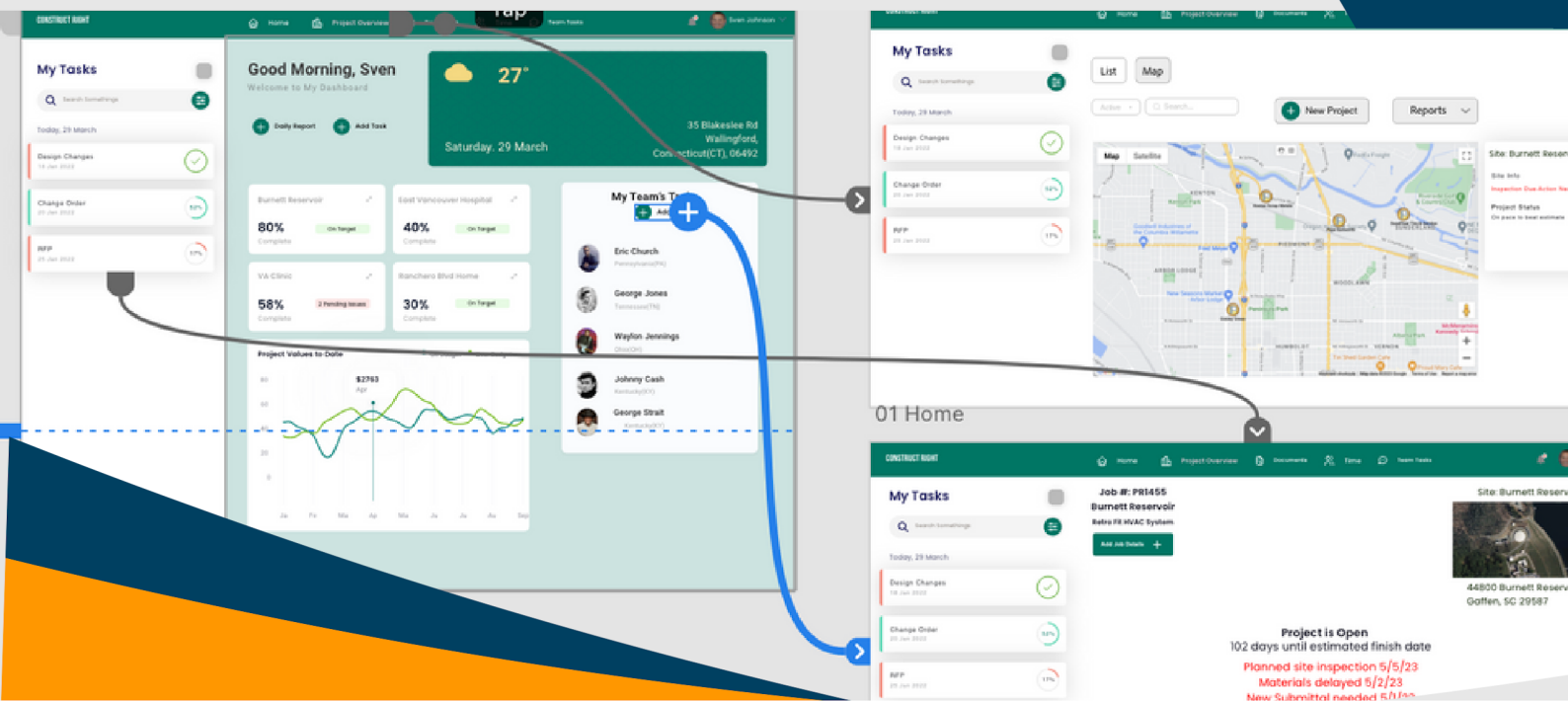
Here are a few examples of functional requirements.

- The system must allow users to log in and out.
- The system must allow users to create, edit and delete jobs.
- The system must send a confirmation email whenever an order is placed.

REQUIREMENTS CON'T

4.3. User Experience Requirements

- **Understand your users.** Who are they? What are their needs? What are their pain points? The more you understand your users, the better you'll be able to design a user experience that meets their needs.
- **Define the goals of the application.** What do you want users to be able to do with the application? Once you know the goals, you can start to design the user experience in a way that helps users achieve those goals.
- **Keep it simple.** The simpler the user experience, the easier it will be for users to learn and use the application. Avoid clutter and unnecessary complexity.
- **Use clear and concise language.** The language you use in the user interface should be clear and concise. Avoid using jargon or technical terms that users may not understand.
- **Use consistent design elements.** The user interface should be consistent throughout the application. This will help users learn the application and feel more comfortable using it.
- **Use feedback and testing.** Get feedback from users as you're designing the user experience. This will help you identify any problems or areas that need improvement.
- **Be flexible.** The user experience should be flexible enough to accommodate changes in the application's requirements or the user's needs.



REQUIREMENTS CON'T

4.4. Architectural Requirements

Architectural requirements are the high-level design decisions that will govern the overall structure and behavior of a software system. They are typically defined in the early stages of the software development process, and they serve as the foundation for all subsequent development work.

There are a number of factors that should be considered when coming up with the architectural requirements for custom software. These include:

- **The business requirements of the system**. What are the specific needs that the system is intended to meet?
- **The technical environment in which the system will be deployed**. What are the hardware and software constraints that the system must adhere to?
- **The non-functional requirements of the system**. These include things like performance, scalability, security, and maintainability.
- **The experience and expertise of the development team.** What are the strengths and weaknesses of the team, and how can they be leveraged to create a successful system?

REQUIREMENTS CON'T

4.4. Architectural Requirements, cont.

Once all of these factors have been considered, the architectural requirements can be defined. This process typically involves the following steps:

- **Identify the key components of the system**. What are the major subsystems that will make up the system?
- **Define the interactions between the components**. How will the components communicate with each other?
- **Choose the appropriate architectural patterns**. There are a number of well-known architectural patterns that can be used to design software systems.
- **Document the architectural requirements**. The architectural requirements should be documented in a way that is clear and concise.
- **The architectural requirements are an essential part of the software development process**. They provide a blueprint for the development team, and they help to ensure that the system meets the needs of the business.

ARCHITECTURAL SURVEY CHECK LIST

Item	Notes
Performance	
Scalability	
Availability	
Maintainability	
Configurability	
Portability	
Functional Usage Monitoring	
Data Manipulation and Transaction Logging and Auditing	
Resilience	
Source Control	
Continuous Integration	
Continuous Deployment	

REQUIREMENTS CON'T

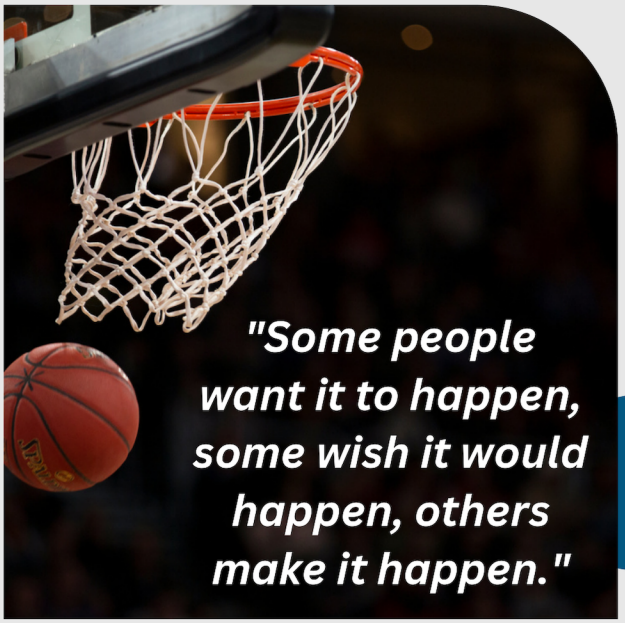
4.5. Security Requirements

- **Identify the assets that need to be protected.** This includes data, systems, and applications.
- **Assess the threats and vulnerabilities that these assets face.** This includes both internal and external threats.
- **Define the security requirements that are needed to protect the assets from these threats.** This includes requirements for authentication, authorization, access control, data encryption, and disaster recovery.
- **Prioritize the security requirements.** Not all security requirements are created equal. Some are more important than others.
- **Document the security requirements.** This will help to ensure that the requirements are understood by everyone involved in the development process.

Here are some additional tips for coming up with security requirements for a custom software application:

- Involve security experts in the requirements gathering process. They can help to identify potential threats and vulnerabilities that may not be obvious to non-security professionals.
- Use a security framework or checklist to guide the requirements gathering process. This will help to ensure that all of the important security requirements are considered.
- Test the security requirements throughout the development process. This will help to ensure that the requirements are implemented correctly and that the application is secure.

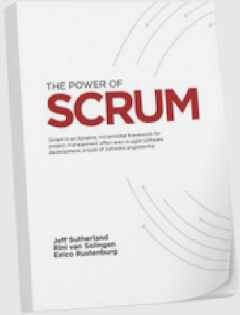
CONCLUSION



Michael Jordan
NBA Star- The GOAT

When embarking on a digital transformation journey, it is important to remember that most projects will encounter detours along the way. It is crucial to embrace flexibility in order to navigate these challenges successfully.

Software development has largely shifted towards an agile approach, which enables leaders to provide input, adapt to changes, and rapidly deploy software to end users. This approach not only reduces overall time and cost but ensures more accurate development and user acceptance of features.



A powerful resource we highly recommend getting your hands on is: *The Power of Scrum* by: Jeffrey Sutherland, D.M. Van Solingen and Eelco Rustenberg