

White Paper

# Top 5 Mistakes to Avoid in IBM i Modernization



looksoftware

A Division of FRESCHÉ LEGACY 

[looksoftware.com](http://looksoftware.com) | [blog.looksoftware.com](http://blog.looksoftware.com) | [info@freschelegacy.com](mailto:info@freschelegacy.com) | [freschelegacy.com](http://freschelegacy.com)

# Contents

Abstract ..... 3

Mistake 1: Boiling the ocean..... 3

Mistake 2: Emulating 5250 on the web ..... 4

Mistake 3: Green screen thinking..... 6

Mistake 4: All or nothing vs. 80/20..... 7

Mistake 5: Forgetting to add business value ..... 8

Summary..... 9



## Abstract

During the last 15 years technology has evolved at an unprecedented rate, and so have the goals & scope of modernization. However, the fundamentals of successful implementations haven't. In this report we examine 5 common mistakes that at best lead to inefficient modernization, and at worst cause failure to implement any new or improved solution.

## Mistake 1: Boiling the ocean

When taking on a new IBM i modernization project, you are typically faced with an existing 5250 application that has a significant amount of functionality and a large number of screens. There is often an incorrect assumption made that the entire application needs to be modernized before delivering anything to the users. Taking on such a large scope in one single attempt often leads to significant inefficiencies and sub-standard results.

It certainly makes good sense to begin with the end in mind, but the smart solution is to take a well thought out iterative and incremental approach. Take a look at what you need to achieve in the long term, but prioritize by measuring the value to the business in the short term, then segment into manageable phases. Typically with front-end modernization projects, getting a basic refaced interface to the users with some simple examples of how it can be made better is a great candidate for the first phase. Getting something out quickly and receiving some feedback is critical to ensure you are on the right track.

Additionally, delivering some simple examples of enhanced functionality that exemplify possible enhancements can help inspire creative suggestions from users. This can be anything that adds value to the application as it stands today and makes the application easier to use or more effective. Getting constructive feedback before going too far, letting you know if the example is high or low value in the real world, or indeed useful at all, can save a significant amount of wasted time and effort. It's often hard for users to describe what is wrong and how to make something better from theory or concepts, but you can foster some valuable insights when there are interactive examples.

Over the last 10 years agile software development has become an increasingly popular approach that works very well with modernization projects. It does require investment in time and effort to implement because it is quite a departure from older approaches but it can yield tremendous benefits. One of the greatest benefits of agile development is that it encourages rapid and flexible response to unforeseen changes and functionality enhancements, which fits in well with today's fast-paced world.

## Recommended Reading

'Agile Software Development' by Robert C. Martin is an excellent introduction to agile development principles. It demonstrates the value that can be created through getting feedback early and often.

## Mistake 2: Emulating 5250 on the web

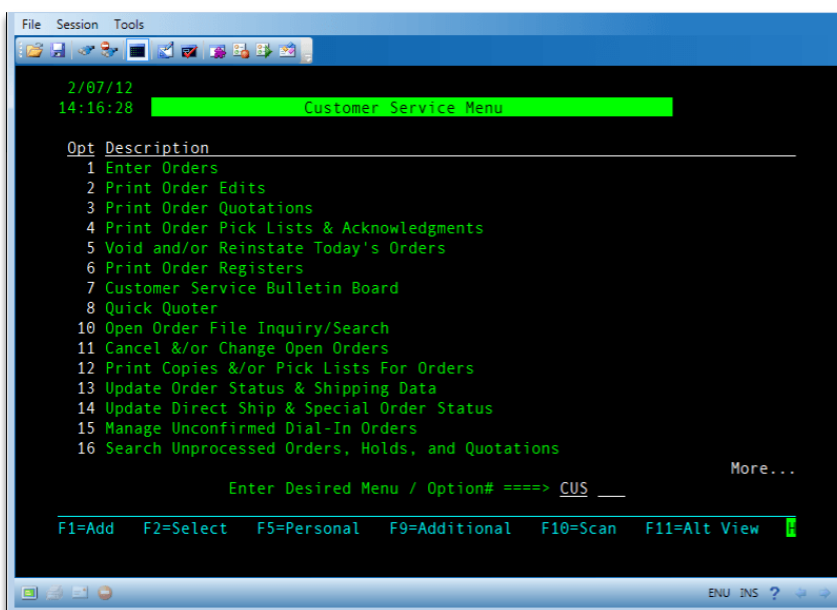
Another common but avoidable mistake is the polar opposite of boiling the ocean. Taking an existing 5250 application and 'flipping a switch' so that it runs on the web without any kind of optimization can actually be a big step backwards in many cases. A couple of commonly used terms used to describe this approach are 'screen scraping' and 'putting lipstick on a pig'. They are negative terms because of the lack of positive results that are gained.

A modern user interface (UI) will be faster to use if it's well built, as you can do many more things than just use the keyboard. You can simplify, shorten, extend, integrate and validate tasks in a way no emulator could. Giving users a fixed pitch 'emulator' font can slow new users down by up to 40%. Modern fonts are easier and faster to read (they are designed that way).

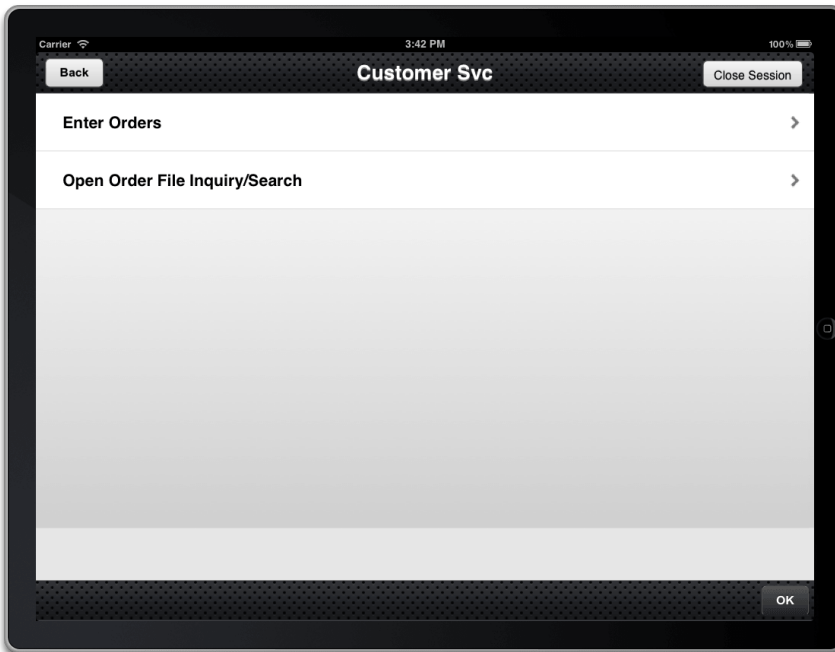
One counter argument to emulating 5250 on the web is that it is easier to deploy to users. That is actually a valid point, but often there is reduced performance and responsiveness if nothing else is done to design the modernized application so that it is an improved experience. Design requires making applications work effectively, fit the platform, and become more intuitive to the task. Developers are often not blessed with the natural ability to create an aesthetically pleasing interface, but the design of modern UIs has become more of a technical skill following well-defined standards. It is something that can be taught if you are willing to learn.

An effective and simple way to get started with some design ideas is to look at popular modern applications that are popular today on PCs, tablets and phones. In particular, look at how navigation can be made more efficient when compared to your existing 5250 applications. If the graphic design tasks worry you, enlist some help from your website team or get a designer in to build you a good standard to work from. They don't have to work on every screen, but can provide global graphics, fonts, styling & a color palette.

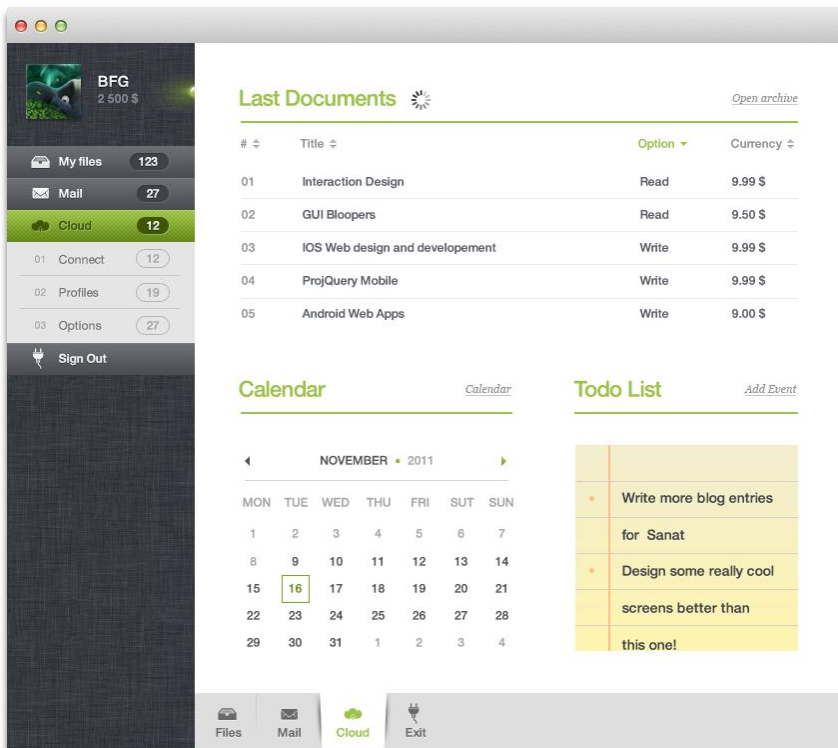
Most importantly, web emulation often lacks any business value, and you will miss a huge opportunity to make your business more productive with happier, quicker and less error-prone users.



**Fig1.**  
5250 version of application.



**Fig2.** Tablet web app - notice only the required items are shown and how simple and easy the app becomes.



**Fig3.** A different app showing truly modern IBM i app style, not simple web emulation.

## Mistake 3: Green screen thinking

The paradigm shift from 5250 application design and navigation to modern graphical applications must be understood before undertaking any kind of modernization project. Fortunately there are some common, specific and practical examples to help you shift your thinking from green screen to a modern graphical UI:

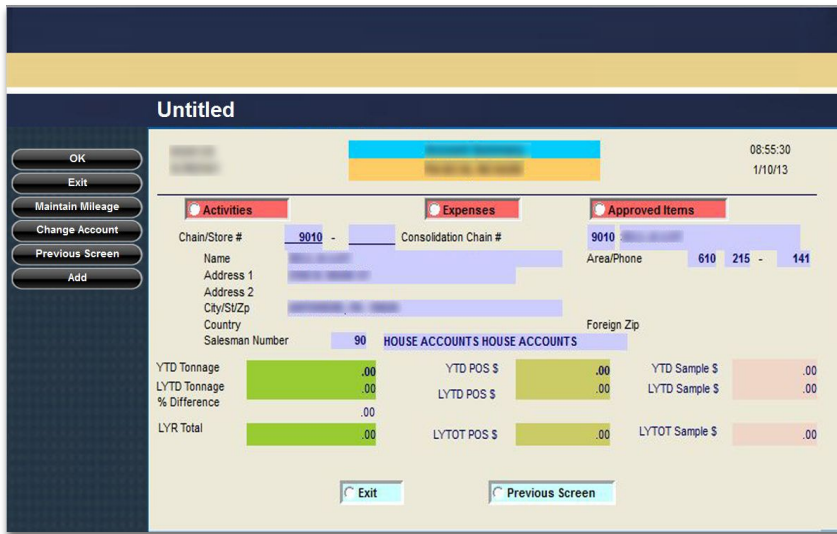
**Error messages.** Here's a typical example from the JD Edwards World application; when a user makes a mistake, as a response the system reverse-highlights the function key area to tell the user to press 'F9=Errors'. The issue is that 'F9=Errors' is not actually shown on the screen at that point. Also, many error messages typically have technical information that doesn't mean anything to the end user. So what should we do? Why not provide the message in a clear and friendly sentence, free from technical terms and jargon so the users don't get frustrated and can resolve their issues without calling support.

**Keyboard maps.** Often developers configure the keyboard to work like Client Access. It's important to look at how using controls that leverage the mouse or touch can make the application more self-evident. The principle is to enable users to work as quickly as possible by embracing the power of the modern device or platform they are using. Making users memorize function keys or keyboard maps defeats the purpose of building applications that leverage the latest technology.

**Technical info on screens.** 5250 applications typically show date, time, virtual device ID, program name and other data that a user does not need. In UI terms this is sometimes referred to as 'computer administrative debris'. For most users this makes the screen look cluttered and takes up valuable space. Your PC, Mac, tablet or phone has a clock, so there's no need to repeat it within the application itself. With regards to technical detail, the modern approach would be a simple popup box showing a description and the required technical information for support, along with the number to call and an option to email it.

**Fixed fonts.** This was highlighted in the previous section, but it's such a common mistake that it's worth repeating again. Proportional fonts (like Arial) are easier to read, look modern and let you fit more data on the screen. Unlike 5250 applications, no modern desktop, web or mobile application uses fixed fonts.

**Using CUA ideas instead of GUI standards.** It's so important to use well thought out standards rather than subjective personal preferences when laying out screens. Learn about standards for application design that have been created by UI experts following much research and testing. Standards provide consistency and enable creation of effective interfaces without reinventing the research and design wheel. Below are a couple of recommended books on the subject of UI design.



**Fig4.** Be careful with color and don't 'roll your own' UI styles.

## Recommended Reading

'Designing The Obvious' by Robert Hoekman. This book is a great place to start with understanding modern design. It's an easy read, and great for the beginner. For more advanced students, 'About Face' by Alan Cooper is a more suitable, if more complex read.

## Mistake 4: All or nothing vs. 80/20

The 80/20 rule stipulates that approximately 80% of your users' time will often be spent in less than 20% of an application. Therefore, it makes sense to spend more time and effort in the 20% that gets heavily accessed than the remainder of the application. A simple example is the sign-on screen. Every user sees it, it is the first impression of your application, and can be very useful beyond just users details such as company news. Also, give some thought to good templates for your menus and then see how you can add real value to the high traffic areas. Your time is valuable so choose to invest it where it will really make a difference.

IBM i developers often design for the 'edge case' – the uncommon or unlikely thing that could still possibly happen. On the other hand, in the PC or web world it's more normal to design for the 'common case'. Whilst this has been the bedrock of the IBM i community's legendary reliability, it can often lead to projects entailing far more work than is actually required. One important example of this is dealing with large numbers of screens present in typical modernization projects. It is unnecessary to individually modernize 100, 1,000 or even 10,000 screens if you set the project up correctly. There are ways to configure global settings and rules to handle the vast majority of cases.

Help and information for users is often neglected as developers shy away from the task of writing help for the entire application. Using the 80/20 principle, it's far better to provide comprehensive help in high traffic areas (step-by-step instructions or a quick 'how to' video can be created quickly), than to have nothing at all. If you can provide 10-20 quick explanations or screen videos for common functions this can dramatically improve the user experience.

## Recommended Reading

'The 80/20 Principle: The Secret To Achieving More With Less' by Richard Koch. Although this book has nothing to with application development or modernization, it can really help you manage your time and help with how to best focus your efforts in your projects.

## Mistake 5: Forgetting to add business value

This is the most important area to focus on when modernizing, yet it often gets forgotten about when it comes time to implement. Even though new devices and interfaces are so exciting and appealing, controls, widgets and graphics offer little value on their own. The starting point for modernization should never be color, device or browser. Look for ways to make the application better support your users and business. It's always good to ask yourself 'what can I do here that an emulator couldn't do?'

Observe how the application is used today. The most effective way to do this is to actually spend time with the end users. If you can truly understand how they are performing their tasks today then you can identify ways to improve them with the modernized application and deliver significant benefits. Find the issues and look for enhancements that will deliver the optimal workflow. This simple step is so often ignored, but it can lead to solutions that reduce errors and process completion times by 40-60%. You can gain significant efficiencies by simplifying common tasks.

Look for areas where integration can remove ALT+TAB switching and re-keying of data. We live in a connected world with a myriad of applications fulfilling a variety of business and personal functions. These apps are all capable of working together, and is a great opportunity to create a seamless experience for the user. If a process requires ERP on IBM i, Word on PC and Google maps on the mobile device, these can all be integrated into a single, consistent UI for the user.

Similarly, where you see users cut and paste info or swap through multiple windows, or even worse, switch sessions on the same IBM i app, basic integration will provide a quick win for everyone. Let the technology take care of data transfer, and you will remove the perception that IBM i is an outdated platform. You will also release users to focus on the tasks where they can add significant value to the business, which is what modernization is all about.



## Summary

If you avoid making these 5 mistakes when modernizing your applications, you will be on track to delivering an experience that will be beneficial to your users and to your business. By taking small steps, learning as you go, and also learning from the mistakes of others, creating a great modern application is much easier than you might think. If you can keep it simple and stay focused on adding business value, you won't go wrong.

Ten years ago modernization was much easier. In those days applications ran on a Windows PC with a 13-15 inch monitor. In 2015 you have a 3-6 inch smartphone and displays in offices and warehouses often greater than 60 inches wide, all with different interactions; touch, keyboard, remote, mouse, stylus etc. The best advice here is to get a sample of the device you are designing for and use it in your own life. Nothing will give you a better understanding. The bottom line is you will struggle to add value in the device arena unless you know what the device can do beyond presenting IBM i info on the screen.

Every application is unique. Every user requirement is unique. And every IT department is unique.

**We invite you to request a free assessment of your application**

Visit <http://go.looksoftware/assessment>

looksoftware is a **division of Fresche Legacy Inc.** Other company, product and service names mentioned herein may be the trademarks or service marks of their respective owners.

The information contained in this document represents the current view of looksoftware on the issues discussed as at the date of publication. Because looksoftware must respond to changing market conditions, it should not be interpreted to be a commitment on the part of looksoftware, and looksoftware cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. looksoftware **and Fresche Legacy** make no warranties, express or implied in this document.