ideaport RIGA

Get results with your enterprise mobile app

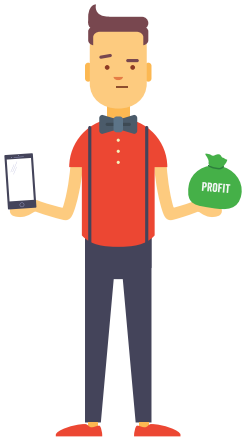
Step-by-step guide to successful enteprise mobility

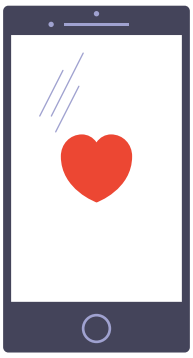
TABLE OF CONTENTS

WHAT MAKES SUCCESSFUL ENTERPRISE MOBILITY?

UNDERSTAND YOUR REASONS AND BUSINESS DRIVERS

Why do you need enterprise mobility? How specifically will it bring profit? You don't need every new trendy thing, unless it delivers business value.





INVEST AND PLAN FOR USER ADOPTION

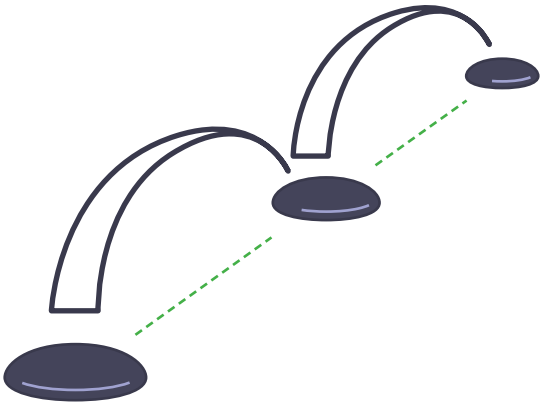
Give it to them, and they'll use it? Not anymore. Your employees had no choice with CRM, ERP or other core systems. But they can and will stick to the old way of coming back to office and typing data into your core system, if the app isn't much better.

Don't misplace your money: without user adoption, enterprise mobility will not bring profit.

ACT AGILE LIKE A STARTUP

What features will make your app work? Nobody knows. If anyone tells you otherwise, they're lying. Every company is different, and the only way to know for sure is to roll out a few features in small sets and act upon user feedback. Like agile startups do.

Beware of 12-month development cycles. Beware of mammoth apps with 136 features. Beware of huge gantt charts with all app releases in the very end. Stay in touch with your users, release frequently and improve in short iterations.



FIND OUT YOUR CHANCES BEFORE STARTING THE JOURNEY

What are your chances to succeed?

5 min checklist

Which phrase ending is closest to your situation?

1. HAVE CLEAR REASONS

"We need enterprise mobile app because..."

- ☐ "...I heard mobile service is becoming more and more popular."
- ☐ "...some employees/ clients request mobile service."
- ☐ "..we can improve productivity of 200 employees who provide service to clients outside the office."

2. ESTIMATE BUSINESS VALUE

"Mobile app will bring profit by..."

- ☐ "...making us known as a forward- thinking company."
- ☐ "...reducing employee downtime, letting clients use more of our services faster, attracting new clients."
- ☐ "...saving 1h/day per employee, meeting the needs of our 3 biggest clients, attracting new big clients."

3. GIVE USERS RATIONAL VALUE

"When opening our app at work..."

- ☐ "...user can access mostly relevant information in read-only mode"
- ☐ "...user can access relevant information and do some tasks he needs to"
- ☐ "...user can access the right information and do all tasks he needs to"

- ☐ "...user must think which device to use for each task."
- ☐ "...user can do most work from any device."
- ☐ "...user can do the same work from any device."

4. GIVE USERS EMOTIONAL VALUE

"When first opening our app..."

- ☐ "...user is not sure how to navigate it."
- ☐ "...user navigates most features easily."
- ☐ "...user navigates it like other familiar apps."

- ☐ "...user feels it is a waste of time."
- ☐ "...user feels it is worth learning."
- ☐ "...user feels more productive."

- ☐ "...user cannot find the task and finds it hard to work on it."
- ☐ "...user finds the task but has some questions about working on it."
- ☐ "...user can find the task started on another device and continue it seamlessly."

Subtotal score: _____ **of 7** (a)

Subtotal score: _____ **of 7** (b)

Subtotal score: _____ **of 7** (c)

5. ITERATE FAST

"We start working on the app development and..."

- ☐ "...list 36 features, make a timeline and pretend to gather user feedback, while actually selling them our plan."
- ☐ "...make a list of 20-30 features and give to users for approval. Adjust the list only if we really have to."
- ☐ "...make a list of 5-10 features and discuss it with users. Remove the features they don't want and add features they request."

- ☐ "...after 6 months we launch the app with 30 features."
- ☐ "...after 3 months we give users the first prototype with 15 features."
- ☐ "...after 1 month we give users the first prototype with 5 features."

6. ADOPT FAST

"After releasing the app..."

- ☐ "...users give us feedback, we need more budget to improve."
- ☐ "...users give us feedback, we improve grudgingly."
- ☐ "...users give us feedback, we improve fast."

7. AND FINALLY

"When I listen to my inner self.."

- ☐ "...I wonder why this project has no potential."
- ☐ "...I hope this project has potential."
- ☐ "...I feel this project has potential."

- ☐ "...I am afraid users stick to old ways and avoid the new process."
- ☐ "...I believe users learn some features and do some work faster."
- ☐ "...I bet users learn the new process and do work faster."

Subtotal score: _____ **of 5** (d)

Subtotal score: _____ **of 5** (e)

Subtotal score: _____ **of 5** (f)

TOTAL SCORE: _____ (a+d) **x 0** **+** **TOTAL SCORE:** _____ (b+e) **x 1** **+** **TOTAL SCORE:** _____ (c+f) **x 2** **=** _____

0 **8** **16** **24**

You better stop here.
Look back at your plan: refine business benefits, high-level use cases, project delivery approach.

Still confused? Let's talk to see if there are better ways to solve your problem.

Proceed with caution.
Check the items with lowest scores and address them early in the project.
Nurture the items with highest scores: they are your power!

Need a hand to fill the blanks?
Let's talk about your project.

Start now!
Keep going with the next worksheets, and let us know if you need tips.

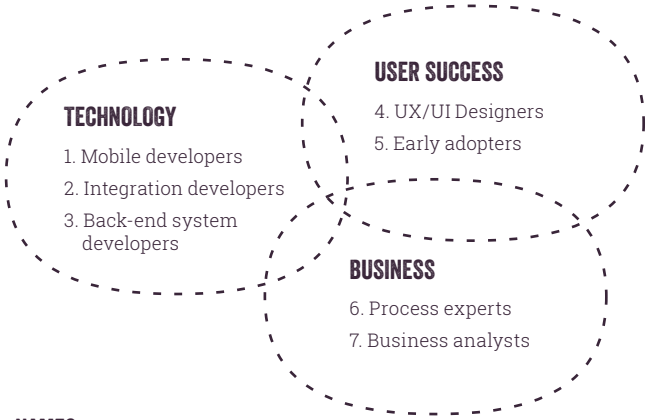
HOW TO BUILD ENTERPRISE MOBILE APP THAT PEOPLE LOVE

Will people love your mobile solution?

8 worksheets to get it right

1. BUILD THE RIGHT TEAM

The right team fills all these roles. Some people may be able to fill in several roles, but make sure you list at least one name for each role:



WHO ARE THEY?	WHAT DO THEY DO?	NAMES
1. Mobile developers	• Code our mobile app	1. _____ 2. _____ 3. _____
2. Integration developers	• Make sure app works with our back-end systems	1. _____ 2. _____ 3. _____
3. Back-end system developers	• Adjust our back-end systems to work with our mobile app	1. _____ 2. _____ 3. _____
4. UX/UI Designers	• Understand user context and needs • Translate this understanding into intuitive navigation flows and appealing design	1. _____ 2. _____ 3. _____
5. Early adopters	• Represent potential app users • Use app's prototype • Provide user feedback	1. _____ 4. _____ 2. _____ 5. _____ 3. _____ 6. _____
6. Business experts	• Know all relevant business processes • Have authority to change them, if needed	1. _____ 4. _____ 2. _____ 5. _____ 3. _____ 6. _____
7. Business analysts	• Translate business requirements and users' needs into functional design/ requirements/user stories	1. _____ 2. _____ 3. _____

Have a specific question?
Schedule a call with one of our professionals: ideaportriga.com/letstalk

2. DEFINE REQUIREMENTS THROUGH JOURNEY MAPPING

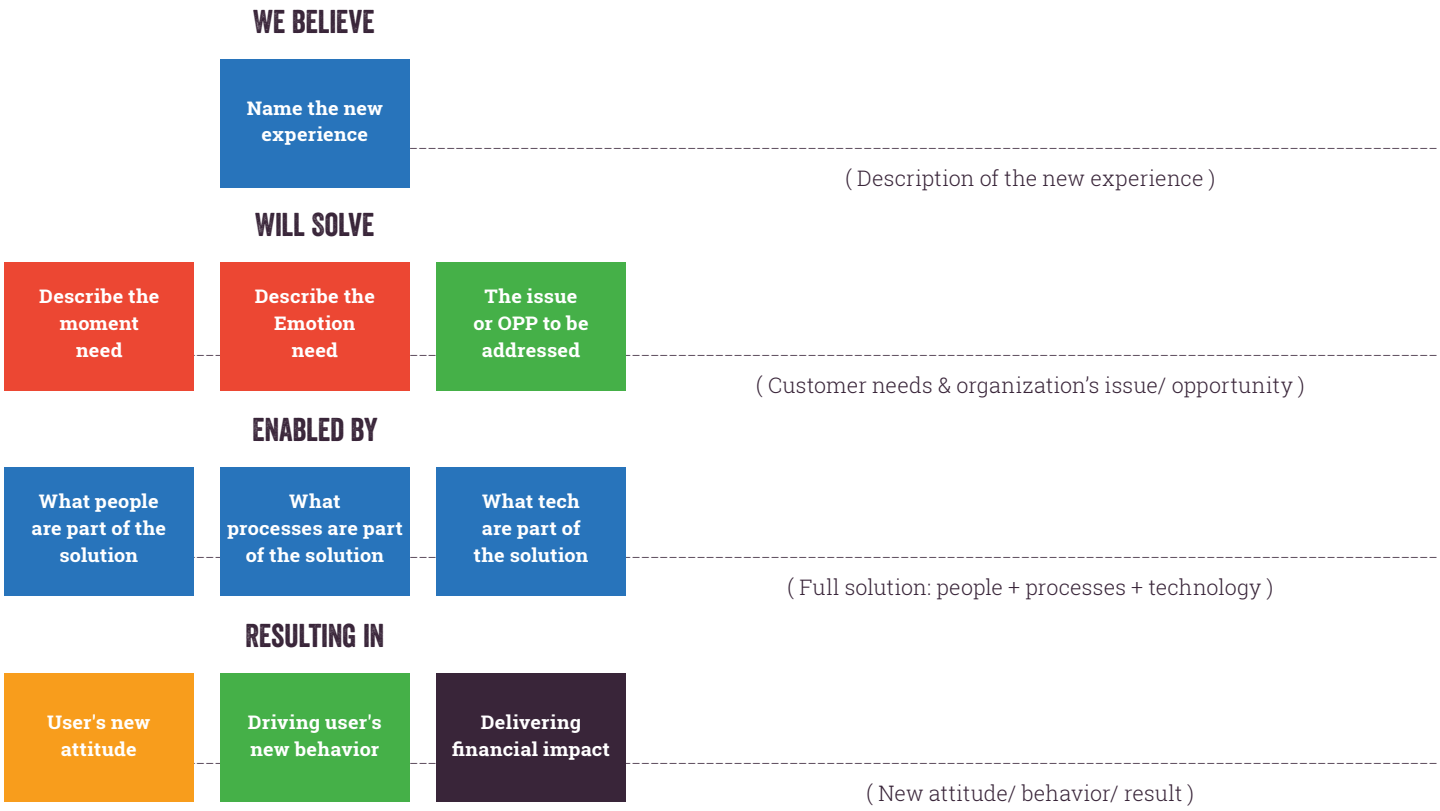
Unlike your core enterprise systems, you cannot force employees to use your mobile app. They can always stick to their old ways: only connecting to your systems in the office. So, let's treat users as customers and win them over.

We suggest user journey maps to complement your usual process diagrams to gather information that the latter often miss.
Journey maps display users' processes, needs, attitudes and impactful moments as they interact with your company.

TOPIC	PROCESS DIAGRAMS	USER JOURNEY MAPS
Main focus	The company	The user
Describes	The company's internal processes, functions, and activities	The process from the user perspective with corresponding company's employees, activities, systems
Terminology	Terms and jargon specific to the company	Plain user language
Appeals to	Logic and numbers	Emotions and attitudes, as well as numbers and financial impact
Who is involved?	Business analysts and business experts	All groups involved in user experience

Research multiple sources on Customer/User Journey Mapping, or just take the approach by Oracle Customer Experience Solutions here:
http://designingcx.com/cx-journey- mapping-toolkit/

At the end of Customer Journey Mapping exercise, you should have several "CX Hypothesis" that melt together both user needs and process improvements. Oracle proposes the following structure to define them:



Have a specific question?
Schedule a call with one of our professionals: ideaportriga.com/letstalk

Whichever way you choose, the minimum viable outcome to move forward should be two lists:

Main users' pain points/needs to address

USER'S PAIN POINT/ NEED	HOW MUCH LOVE WILL THE SOLUTION BRING ♥♥♥ ♥♥ ♥	HOW THE APP WILL HELP USERS
1.	☐ ☐ ☐	1.
2.	☐ ☐ ☐	2.
3.	☐ ☐ ☐	3.
4.	☐ ☐ ☐	4.
5.	☐ ☐ ☐	5.
6.	☐ ☐ ☐	6.

Main process improvement ideas to boost business value

PROCESS IMPROVEMENT	HOW MUCH PROFIT WILL THE SOLUTION BRING € € € € € €	HOW THE APP WILL BRING PROFIT
1.	☐ ☐ ☐	1.
2.	☐ ☐ ☐	2.
3.	☐ ☐ ☐	3.
4.	☐ ☐ ☐	4.
5.	☐ ☐ ☐	5.
6.	☐ ☐ ☐	6.

3. MANAGE USER DEVICES AND APPS

Before building the app for your employees, you need to understand how you will handle their mobile devices and apps.

Get the answers to the following essential questions:

1. What mobile OS will our app run on?	☐ iOS	☐ Android	☐ Windows
2. What form factors user devices will have:	☐ Smartphone	☐ Mini tablet	☐ Tablet
3. Who provides the devices?	☐ Company, business use only	☐ Company, personal use allowed	☐ Employees bring their own

What are the brand/model/OS versions of primary target mobile devices:

☐ _____	☐ _____	☐ _____
☐ _____	☐ _____	☐ _____
☐ _____	☐ _____	☐ _____
☐ _____	☐ _____	☐ _____

5. How will users install the app?

_____

6. How do we know which app version is in use by each employee?

_____

7. How will we know which employee is using which device(-s)?

- ☐ By IMEI
- ☐ By phone number
- ☐ By app login
- ☐ By corporate account
- ☐ By social account
- ☐ Other way (specify):

  _____

8. How will we wipe data from stolen, lost and former employees' devices?

- ☐ Built-in app functionality
- ☐ Part of centralized mobile device management (MDM) solution
- ☐ Part of centralized mobile application management (MAM) solution
- ☐ Other (specify):

  _____

9. What other MDM and MAM features do we need?

☐ _____	☐ _____	☐ _____
☐ _____	☐ _____	☐ _____

4. MAKE EXTERNAL CONNECTION SECURE

For 20 years you've been running enterprise applications from on-premise servers, talking to each other over secure internal networks. Mobile environment has very different security standards.

Here are most common security concerns to address:

1. How do we verify app user's identity?

- ☐ Fingerprint
- ☐ Login & password
- ☐ Corporate account
- ☐ Social account
- ☐ Personal certificate
- ☐ Other (specify):

_____

2. Is data stored on user's device (-s) encrypted? ☐ No ☐ Yes Comments _____

3. Is data sent to and from user's device (-s) encrypted? ☐ No ☐ Yes Comments _____

4. Are back-end systems' APIs directly exposed in calls made by mobile apps? ☐ No ☐ Yes Comments _____

5. How do we verify if API call is made by our app?

- ☐ Application key / token
- ☐ Session key / token
- ☐ Token
- ☐ Other (specify):

_____

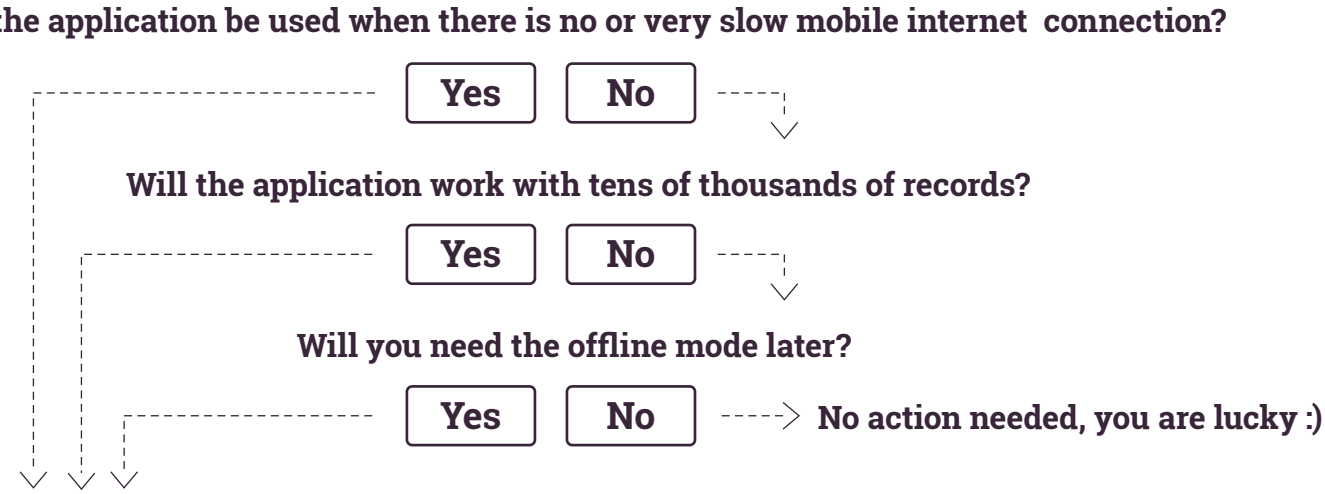
6. What else might compromise the security of our mobile solution?

RISKS	LIKELIHOOD	SEVERITY	SOLUTION
1.	☐ ☐ ☐ (High Low)	☐ ☐ ☐ (High Low)	1.
2.	☐ ☐ ☐	☐ ☐ ☐	2.
3.	☐ ☐ ☐	☐ ☐ ☐	3.
4.	☐ ☐ ☐	☐ ☐ ☐	4.
5.	☐ ☐ ☐	☐ ☐ ☐	5.

Have a specific question?
Schedule a call with one of our professionals: ideaportriga.com/letstalk

5. DESIGN FOR OFFLINE

You need your user do get this job done well with bad, terrible and no internet connection. Or do you?
Here's how to decide:



Design for offline mode:

1. Data will be stored locally in

a. SQL Lite DB b. Key-value store c. Other: _____

Comments: _____

2. Synchronizing the data

Object	Static, Slowing changing or Real-Time	Number of records in sync call	Number of records stored locally	Conflict resolution strategy

3. Describe how the offline mode will work

_____

_____

_____

_____

6. SPEED UP DATA EXCHANGE

If your app user only spends 5-7 min per interaction, you do not want him looking at “loading” screen for 30 seconds. Ideally, the data should be fetched instantly, in "< 1 second". How do you achieve it, with 2-5 integration layers each adding tens or hundreds of milliseconds?

1. Understand what data mobile app should fetch through the middle tier API and how each API is resolved to one or several back-end systems.

DATA OBJECT	OPERATION	MIDDLE TIER API	INTEGRATION FLOW	BACK-END SYSTEM’S INTERFACE

2. Consider the following strategies to improve performance for each of the data exchange scenarios.

Mobile App

- [ ] Implement local storage

Communication Tier

- [ ] Sync data in the background
- [ ] Shorten the messages’ format
- [ ] Compress the messages as you send them
- [ ] Implement asynchronous patterns
- [ ] Do parallel calls

Middle Tier

- [ ] Cache the data

Integration Flows

- [ ] Do parallel calls
- [ ] Implement transformation logic
- [ ] Implement business logic
- [ ] Orchestrate back-end service calls

Back-end Systems

- [ ] Discuss your options to improve back-end systems' interface performance

7. ROLL OUT AND ITERATE

Finished your first app's release party? Time to focus on your users now.

1. **Roll out the app across different organizational units**

With users from several organizational units (e.g. geographical units), it makes sense to roll out your app gradually, at least with the first three user groups. While at it, get as much qualitative user feedback as possible: make sure the app really solves their pains.

CALENDAR	GROUP/ UNIT 1	GROUP/ UNIT 2	GROUP/ UNIT 3	REST OF GROUPS/ UNITS
Local contact person				
Demo date				
Training date				
App ready to install				
Launch date				
1st feedback session date				
2nd feedback session date				
3rd feedback session date				

2. **Make data-informed decisions**

Combine the insight from qualitative user feedback with quantitative data from your mobile solution's analytical component on app usage.

What to look for in your analytical module:

- How is the number of users increasing?
- How is time spent with the app changing?
- Which features stuck with the users and which did not?
- Is application response time remaining satisfactory?
- Does the app crash too much?

3. **Plan new release features**

Now, equipped with users feedback and hard analytics data, return to your user journey map (see page 7) and update it.

FEATURES TO IMPROVE	FEATURES TO ADD	FEATURES TO DROP
1.	1.	1.
2.	2.	2.
3.	3.	3.
4.	4.	4.
5.	5.	5.

8. PROVIDE EFFECTIVE SUPPORT

There will be rainy days for your mobile solution. Nasty defects will get into the latest app's release, the middle tier will be running slower than usual, integration with back-end will throw exceptions, and back-end systems will sometimes need maintenance, too.

You must be there for your users when this happens and provide genuine support. For that:

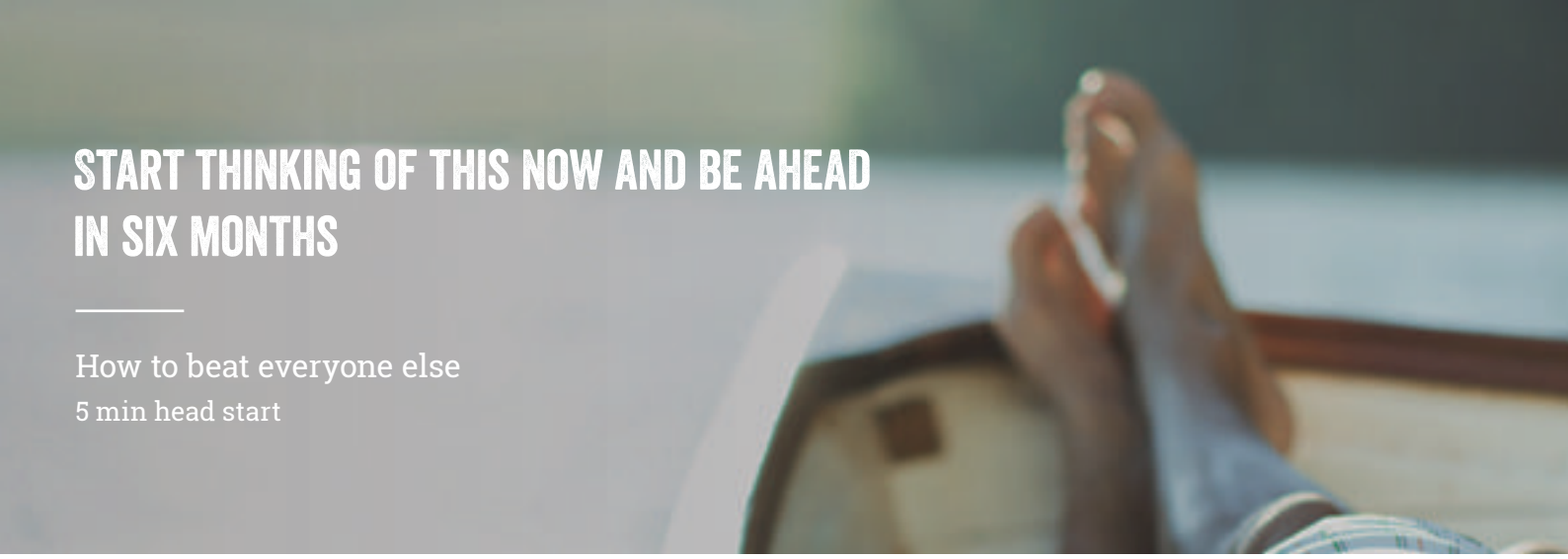
1. **Provide your support staff with relevant information to do their job well :**

AREA	SOURCE OF INFORMATION
User's access and setup	
User's device and app version	
App's crashes	
Middle tier's infrastructure status	
Errors in integration flows	
Performance of back-end systems' APIs	
Availability of back-end systems	
Planned deployments of new app's version	
Planned maintenance activities on the back-end systems	

2. **Build a list of common issues that may happen to your mobile solution. Analyze them, share findings with your support team, then think how to minimize their impact and likelihood:**

TYPICAL ISSUE	HOW USERS EXPERIENCE IT	HOW SUPPORT DETECTS IT	WHAT IS THE ROOT CAUSE	HOW TO RESOLVE THE ISSUE	ROOT-CAUSE MITIGATION PLAN
Area: User Setup					
Area: Mobile App					
Area: Middle Tier					
Area: Back-end systems					

3. **Consider sharing relevant support information with end users. Some of them will use it to help themselves and unload your support staff to focus on more serious user problems and proactive maintenance.**

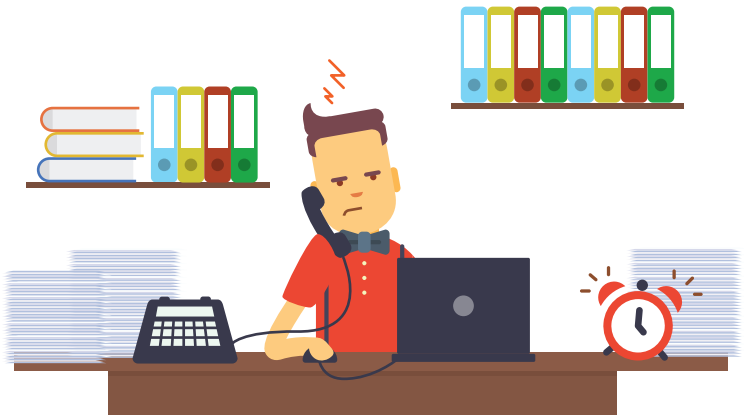


START THINKING OF THIS NOW AND BE AHEAD IN SIX MONTHS

How to beat everyone else
5 min head start

Most of your competitors stop here. However, if you want to be ahead of everyone else when you need it most, invest a few minutes now to think and act on these key issues:

1. SCALABILITY OF YOUR MOBILE SOLUTIONS



Mobile solutions usually start small and steadily add new users and features. This means more mobile apps working with more back-end systems, each providing a wider set of services.

It is easy to think you can always worry about scalability later, when the time calls. However, time tends to call in the middle of a heated project, when a key employee is on vacation and your C-level boss shows up, seriously concerned about the situation.

Think: **My basic mobile app is up and running. Let's see how I can scale it in the future:**

Act:

1. ☐ Prepare and run performance tests: know your limit.
2. ☐ Estimate the upper load level that your current mobile solution setup can handle.
3. ☐ Define how you can horizontally scale the solution.
4. ☐ Run experiments to determine your scalability plan and start the work on plan's execution.
5. ☐ At the same time, have mobile solution performance monitoring in place, analyze the actual performance continuously and adjust your scalability plan accordingly.

2. MOBILE BACKEND SERVICE

At first, your mobile app most likely will run on top of API provided by back-end systems. However, as your app scales, you will need faster data transfers, robust offline support, and more sophisticated data synchronization mechanisms.

If you are already using one of Mobile Backend Service / Mobile Backend as a Service solutions, you are well prepared. If not, it's time to consider one.

Typical features of a Mobile Backend Service solution:

1. Secure access
2. User management
3. Analytics
4. Database with consolidated back-end systems' data
5. Offline sync
6. Push notifications

Think: **Should Mobile Backend Service be part of my mobile solution?**

Act:

- [ ] 1. Discuss this topic with your enterprise IT architects.
- [ ] 2. Evaluate what current mobile solution challenges Mobile Backend Service can resolve.
- [ ] 3. Research different solutions, talk to other companies, engage vendors.
- [ ] 4. If you decide to go for the Mobile Backend Service solution, include it in your mobile app project plan.

3. CENTRALIZED API MANAGEMENT

As the functionality of the mobile app expands, you need a comprehensive API management capability in the middle tier. API management also brings value when you build other mobile apps, create role-specific web apps or expose your systems' API to partner companies or the public. Avoid rebuilding the same middle tier functionality in every solution: implement a centralized API management solution instead.

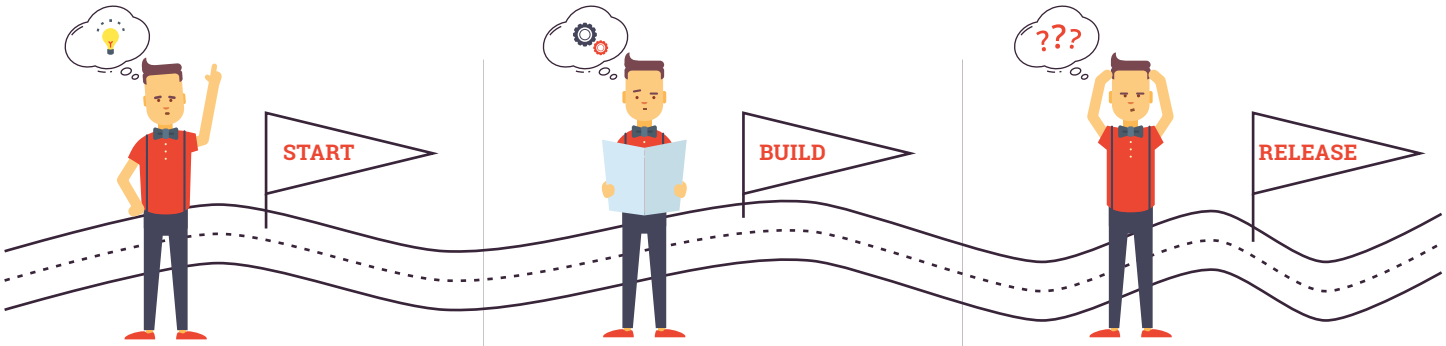
Typical features of an API management solution:

1. Secure API gateway
2. Managing API of back-end systems
3. Managing API's consuming apps and users
4. API calls' metrics and events' logs
5. Developer portal with API documentation, sandbox, run-time analytics

Think: **What applications do I have now? What applications have I planned? Which of them will need access to my internal systems' API? How to make it convenient, secure and scalable? Will my organization benefit from an API management solution?**

Act:

- [ ] 1. Evaluate candidate applications to use centralized API management solution.
- [ ] 2. Find out how do these applications deliver the functionality mentioned above.
- [ ] 3. Understand what functionality are they missing.
- [ ] 4. Evaluate how much would the API management solution benefit these applications.
- [ ] 5. If you find that there might be a business case for API management in your organization, research solutions, talk to other companies, engage vendors.



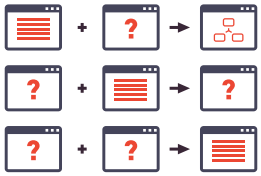
Planning your new enterprise mobile app project?

ideaport RIGA

We can define how the app will make your company and mobile employees happier.

Let's talk

Need a solution design and start coding the app?



ideaport RIGA

We help you pick and choose, design and build all vital mobile solution's elements.

Let's talk

About to go in production, but worried about a few support questions?

ideaport RIGA

We help you track app performance and assist your users.

Let's talk

We know what we do: We have launched several enterprise apps across several industries and geographies over the last seven years. We have learned the differences between mobile solutions that fail and those that make headlines first-hand, and we are here to share them with you.

Optimal tech: We use Xamarin or JavaScript + Cordova for mobile app development. These technologies ensure optimal usability while keeping a single code base, thus saving your budget.

We know common pitfalls: Mobile projects often get too focused on building the app itself, neglecting the middle tier. It's a shame. Middle tier with security, fast data syncs and a wide set of exposed back-end systems' API is key to make your app feel seamless: something your users would love to work with.

See the full picture: Your app isn't done after user acceptance testing and release. It just starts here. Poor user support often destroys the adoption of the most beautiful and functional enterprise apps. We prevent that. Your support teams get built-in features and analytics to see what users are doing in the app, what crashes the app, how middle tier handles API calls, and how fast the back-end systems respond.