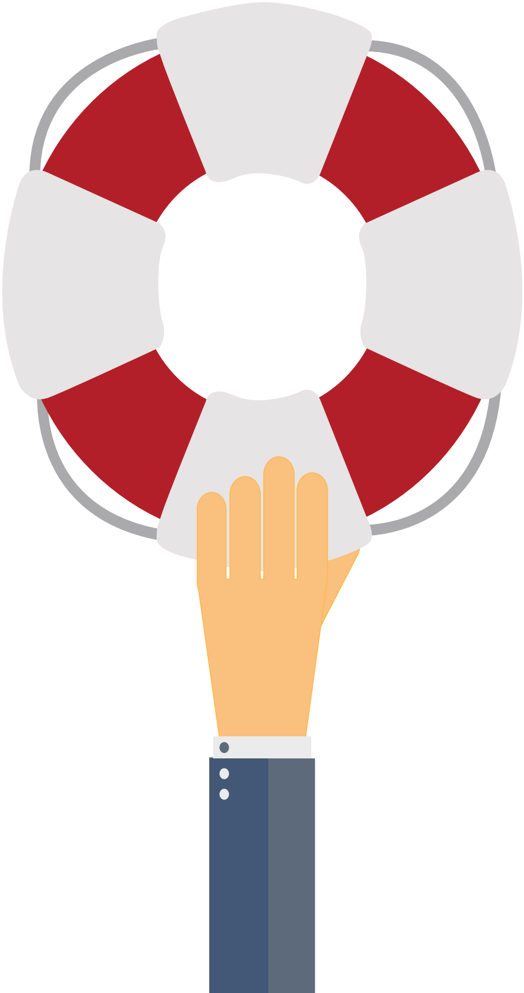
IN-APP SUPPORT: THE COMPLETE GUIDE

REVTWO

CONTENTS



THE MOBILE APP MARKET

In a hyper-connected world, we rely on mobile apps to get things done, keep us organized and provide entertainment. In fact, Statista predicts that there will be more than 268 billion mobile app downloads by 2017. There's no question apps will continue to play a major role in our lives for years to come – and represent a massive business opportunity for publishers, companies and developers.

But profiting on mobile apps is easier said than done. Beyond creating a great product, to survive and profit publishers have to simultaneously deliver a terrific user experience, generate strong app store reviews, grow the user base and limit app abandonment – all on a pretty limited budget.

Making success even more difficult: the mobile app market is incredibly unforgiving. If publishers come up short in any of these areas, users will stop using your app, delete and move on to the next best thing.

This e-book provides critical strategies and essential considerations for app publishers **looking to grow their user base and create a top-notch experience.**

WHY IN-APP SUPPORT MATTERS

Let's face it, many times the last thing we are thinking about when we are developing our apps is support. Our focus is on the app's mission, usability, feature set, and look and feel. Unfortunately, this lack of foresight often leads to catastrophic results.

As any good growth manager will attest, what happens after deployment is the most significant part of an app's success. Consider the following: Your app is released, it is featured by Apple and Google on their respective stores, it rockets to the top of the download charts. Initial usage numbers are terrific, then reality hits. Usage numbers start to dwindle, app ratings and reviews reflect challenges that are unmet. New customers keep coming in, but they are having a hard time making up for the losses. You find yourself on a treadmill. Unfortunately this is an all not uncommon scenario. In fact, @andrewchen reports that the average app loses its entire user base within a few months.

> “The average app loses its entire user base within a few months” - @andrewchen

GET YOUR FACTS STRAIGHT

82% stopped doing business with a company due to a bad experience

16% will buy from a competitor if they encounter a problem

95% of customers who have a bad experience share it

25% mobile apps are abandoned after a single use

In the ultra competitive world of apps, there is always another app waiting in the wings to to take your customers. **In-app support is often your only line of defense** against a bad review and app abandonment. Without it your customers are left to fend for themselves with their only outlet being one star ratings and deleting the app.

OPPORTUNITY PRESENTED

While the negative consequences to app adoption and traction can be quite stark, there is also a positive aspect of in-app support that is often overlooked. For many types of apps there is a direct transactional correlation between support and economic outcome.

APP-BASED E-COMMERCE

A common area where in-app support drives top-line success is app-based e-commerce. Leading apps have introduced concierge services for their VIP customers where personal shoppers support app based shopping experiences. In these apps, a customer can request a product expert and be connected with company resources for a personalized co-shopping experience. During the co-shopping session, the personal shopper is able to see the customer's screen, have a conversation with them, and send them product suggestions in the form of deep-links.

IOT PRODUCTS WITH COMPANION APPS

Consider an Internet of Things (IoT) product with its companion app. Connecting a product to the app on the customer's wifi network, especially when the product itself may not have a user interface, can be extremely challenging. A hiccup during setup can result in significant top line impact with in-app support being the only thing standing between that company and a multi-hundred dollar product return.

TOP CHALLENGES AND OBJECTIVES

Many apps operate in an environment where they can touch millions of users. In that type of high volume environment, it is often not practical or potentially even economically feasible to interact with users on a one-on-one basis. When putting your app support strategy together, you will need to factor in the cost of response vs. value of user to determine the types of resources to bring to bear. For example, high value customers can justify high value responses, while low value customers warrant lower cost solutions. All of these factors must be considered as you design your in-app support response.

Before you start your In-App support program it is important to align your execution with your program's goals. Typical best practice goals include:

ANSWER QUESTIONS WITHOUT HUMAN INTERVENTION

Many successful apps share the challenge of overwhelmed support agents. We have seen some apps with ticket volumes approaching double digit new questions per agent per minute. There is only one real solution to handling this type of ticket volume and that is to answer the questions before they hit the help desk. We believe that It should be your **goal to handle 70% of your support questions in this manner.**

MAXIMIZE AGENT EFFICIENCY

Even with 70% of app questions handled, the remaining 30% can still represent a significant volume and a significant challenge for your help desk. Your agents need to be equipped with interfaces and tools that allow them to handle the maximum number of tickets efficiently and accurately. An agent should be able to work on multiple tickets simultaneously, should be able to find and deliver relevant FAQs quickly, and should have a useful set of troubleshooting tools at their disposal to speed ticket resolution.

ENGAGE YOUR CUSTOMER BASE

The best asset of your app isn't the code or special algorithms, it's the app's customer base. Utilizing a community support mechanism that engages the user base and enables them to communicate with each other, answer each other's questions, and learn the nuances of your app's subject matter can be the most effective weapon in fighting user churn. Apps with engaged communities become the subject matter experts on their topics and, as a result, are incredibly sticky.

SOLVE HIGH IMPACT PROBLEMS

Not all problems are created equal. While it is important to handle all challenges, many types of problems can lead to significant business impact. Those high impact problems fall into two distinct categories:

- **a.** Problems that have an inherent economic impact - i.e. failed transactions, risk of product returns, and the like or
- **b.** Problems that impact high value customers.

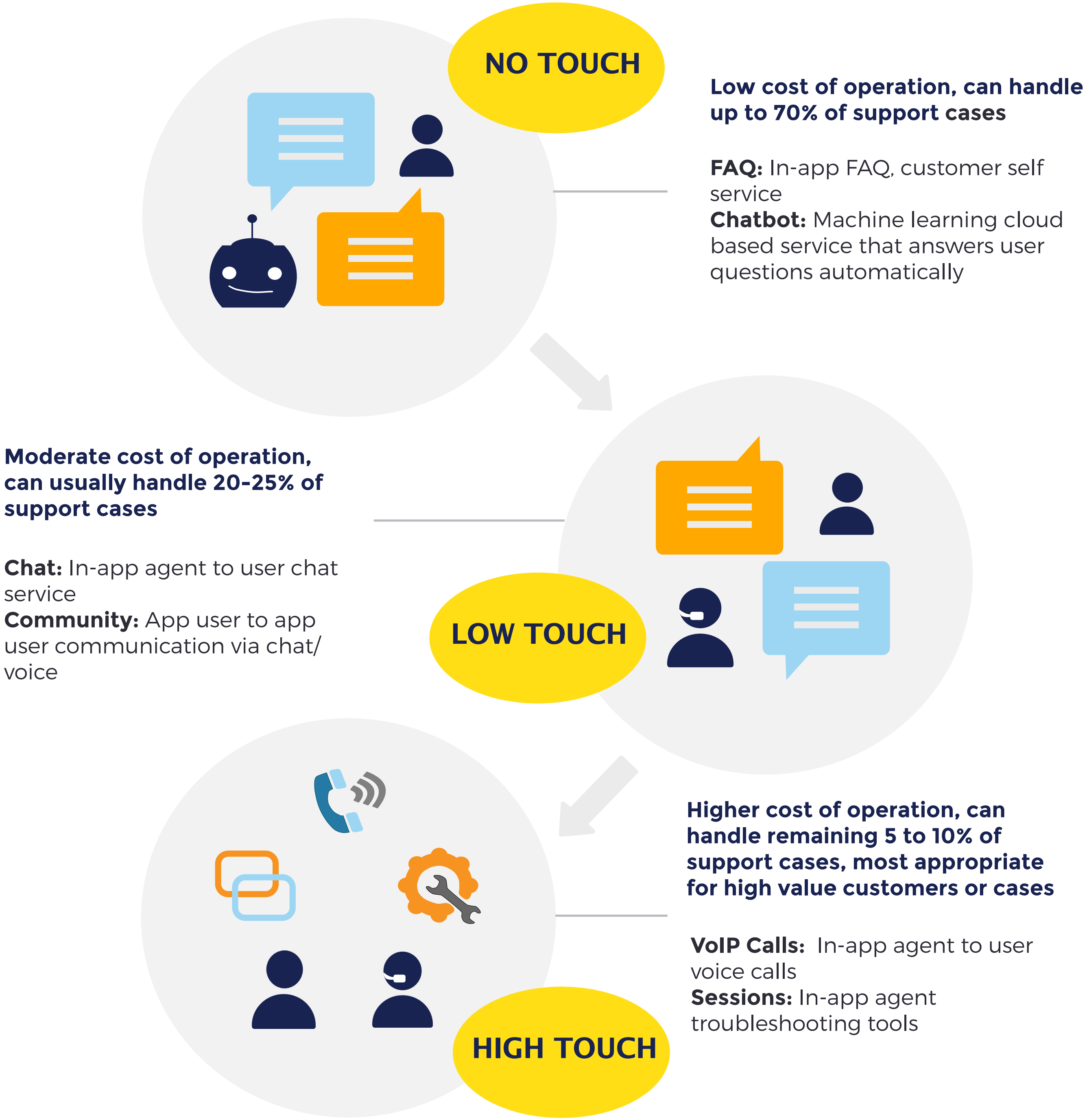
It is important that your in-app support strategy provides an escalation path for these types of situations.

CAPITALIZE ON HIGH VALUE OPPORTUNITIES

As we previously mentioned, support is not just about problem resolution, it is also about leveraging your staff to form a deep, long lasting relationship with your customers. Many apps and products have inherent opportunities for value creation. When considering your in-app support strategy, don't get so focused on ticket resolution that you miss the vein of gold at your feet.

IN-APP SUPPORT STACK

A rich in-app support solution stack has many parts that work both **in conjunction** with each other and **independently** to provide a comprehensive support system.



NO TOUCH

Low cost of operation, can handle up to 70% of support cases

FAQ: In-app FAQ, customer self service
Chatbot: Machine learning cloud based service that answers user questions automatically

LOW TOUCH

Moderate cost of operation, can usually handle 20-25% of support cases

Chat: In-app agent to user chat service
Community: App user to app user communication via chat/ voice

HIGH TOUCH

Higher cost of operation, can handle remaining 5 to 10% of support cases, most appropriate for high value customers or cases

VoIP Calls: In-app agent to user voice calls
Sessions: In-app agent troubleshooting tools

MACHINE LEARNING CHAT BOTS

OVERVIEW

Machine Learning Chat BOTs are the over achieving younger sibling of the FAQ. They accomplish the task of resolving a high quantity of tickets at low cost and with little management, but they do so much more effectively and with less overhead to the user.

BOTs can field questions from multiple sources, such as in-app questions, emails to a support line, or web queries. As a result, putting a BOT online is very simple and can often be retrofitted into the support process without a revision to the app itself.

Furthermore, BOTs become better as they work. Consider the following, a BOT receives a question from a user, it reviews the question, determines that there are three likely answers, then returns the best answer or all three to the user. The user reviews the answers and determines that answer #2 is the one that resolves their current challenge and feeds that information back to the BOT. The BOT leverages this feedback and updates the strength of this answer for this question allowing it to give a better answer next time.

KEY FEATURES

1) Answer authoring environment plus import capability

2) Entity based question matching

3) Machine learning feedback

4) Help desk and user integration / feedback capability

5) Can source from multiple question streams, email or REST

BOTs run 24 hours a day, 7 days a week, and don't take holidays off. The more knowledge you pour into a BOT, the more it runs, the better it gets. BOTs are not limited by answer volume in the way an FAQ is. More answers just enable the BOT to handle more questions. A well designed BOT solution can be plugged in and can yield up to 70% response rate almost immediately.

PROS AND CONS OF CHAT BOTS

+	-
Can be retrofitted into existing apps without code changes	BOT answers need to be identified and authored
Simple to author	
Learns as it runs	
Can handle up to 70% of ticket volume	
Can handle a high volume of answers	

CONVERSATIONAL CHAT BOTS

Conversational Chat BOTs have the goal of interacting with users as naturally as possible. This style of BOT combines natural language processing with knowledge response. The additional complexity of human imitation combined with the back and forth nature of human conversation currently limits their applicability to a limited set of tasks where they can be programmed carefully to yield a satisfactory result.

IN-APP FAQ'S

OVERVIEW

App FAQs (Frequently Asked Questions) are the lowest form of support available for apps. Typically, FAQs are presented as a hierarchy of categories, where each category has a set common question, followed by a detailed answer. FAQs require users to browse through categories and then to compare the questions in that category to their own. When the user finds a question that matches their query, they read the answer and then determine if the answer satisfies their need. The biggest challenge of the FAQ is that that it depends, both on the patience of and technical acumen of the person with the current question. Additionally, the deeper the hierarchy gets, the more burden it puts on the user - so FAQs have a natural limit to number of questions they can handle.

KEY FEATURES

1) FAQ authoring capability

2) FAQ migration tools

3) FAQ category browsing and search

4) Help Desk to FAQ integration enabling help desk to search FAQs

5) Ability to push FAQ changes to app

PROS AND CONS OF FAQ'S

+	-
Limited help desk involvement	Time consuming for user
Simple to author	Ineffective for deep hierarchies
Reduces help desk load	Doesn't learn from user interactions
	Built into app
	FAQ needs to be identified and authored

IN-APP CHAT

OVERVIEW

Chat-based support systems provide bi-directional communication between agents/ BOTs and a user. Chat is fast, light and does not require immediate responses. Additionally, it is the most popular form of communication for millennials. Best practice support systems prompt users to start their chat in the form of a question which can then be routed for automated handling.

A chat that is not handled by the automated response system, is entered into the help desk as a support ticket. When the agent takes the ticket, the chat process begins.

KEY FEATURES

1) Standard chat interface

2) New message notifications

3) Rich media types

4) High volume agent interface, suitable for multiple chat conversations at a time

5) Ticket system integration

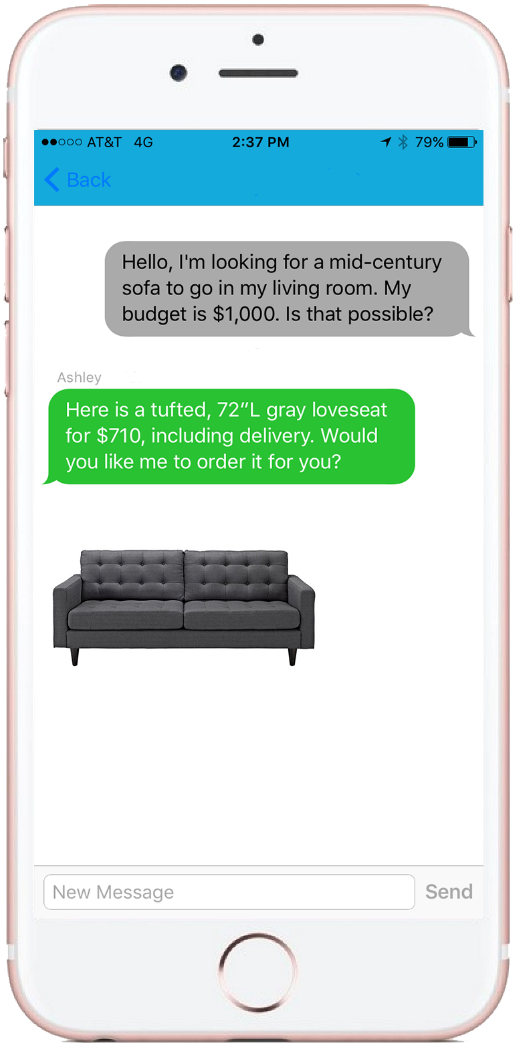
A chat based support system will typically have the following capabilities:

1) Ability to embed rich information in the chat itself, such as

- Pictures
- Links
- Deep links that the user can click to change app screens

2) Specialty agent features, such as

- Ability for agent to handle multiple chats at same time and from the same screen
- Ability to look up answers for common questions and embed them in chats
- Agent feedback for automated answer systems



Hello, I'm looking for a mid-century sofa to go in my living room. My budget is $1,000. Is that possible?

Ashley

Here is a tufted, 72"L gray loveseat for $710, including delivery. Would you like me to order it for you?

Typical support chat interface

PROS AND CONS OF IN-APP CHAT

+	-
Users like to chat, easy to do, not personal	Not super fast
Agents can multiplex between customers	Has limited context

IN-APP CALLS/SESSIONS

OVERVIEW

When capitalizing on high value opportunities or working on high impact problems, there is no substitute for the high bandwidth communications of a voice call. In-app calls work differently from typical phone calls to an 800 number in a couple of critical ways. In-app calls usually work on a call-back basis. A request for assistance is placed by the user from inside the app which is then routed to the help desk. When the help desk picks up the request, they initiate a VoIP call back to the user.

A typical workflow should look like this:

1) App user receives an in-app or out-of-app notification
2) Selecting the notification routes the user to an answer screen within the app, similar to skype
3) The user answers the call from inside the app
4) Audio channel is open

KEY FEATURES

1) A functional VoIP interface embedded in the app

2) Call management tools (mute, hangup, speaker)

3) New call notifications

4) Ticket system integration

5) Troubleshooting tools available during sessions

(In-app calls often start life as a question that routes initially through the chatbot and agent chat support responses.)

Because VoIP calls utilize Internet Protocol to communicate, it is possible to communicate additional information over the same channel during the call session. In these systems a data channel is established in addition to the voice channel that provides the foundation of tool based interactions.

Typical capability for these session based tools are:

1) Screen Sharing:
Screen sharing has been a common tool in PC based application support for many years and in-app screen sharing provides similar benefits. Mobile apps run in a sandbox, therefore they only display the screen of the app itself and their security model prevents agents from performing input.

Additionally, many of these tools will provide a highlighter that the agent can move to provide the app user with a visual indicator of where to click.

2) Deep Linking Content Sharing

Often agents want to share information with an app user. This is especially important for collaborative sessions for e-commerce apps. Deep Linking provides a means for an agent to send information such as product suggections to an app user. The link can be used to navigate to a specific page in the app. For example, if an agent wants to recommend a pair of boots to a user, they can send a link to the customer that will take them to the product page for those boots in the app.

3) Troubleshooting Tools

Troubleshooting tools are a portfolio of tools that are used to find and diagnose problems. Examples of such tools include:

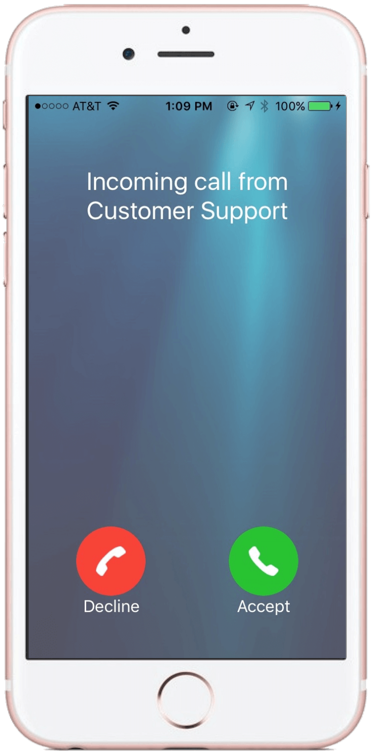
Logs: Many apps generate log messages when things go wrong. Being able to see those messages, historically, or in real-time as users change screens or go about normal app activity can be an extremely valuable troubleshooting aid.

System Information: Apps run in the environment of a phone or tablet, which by its very nature is a limited resource platform. Being able to view information in the system settings can speed diagnosis of problems and identify if underlying resource problems are resulting in bad app behavior.

Database Access: Many apps are built with an embedded SQLite database. SQLite is used to store information that the users enter. For many types of issues, being able to view that information can be of very high value.

File Access: Similar to the database, many apps have local files, such as pictures, audio clips, or the like. The ability for agents to upload or download those files can be of significant value to a support agent.

While logs and system information access can be useful to agents, file and database access should be reserved for developer level support.



Incoming call from
Customer Support

Typical In-App call answer screen

PROS AND CONS OF IN-APP CALLS/SESSIONS

+	-
Ultimate in-app troubleshooting	One-to-one agent to issue
Voice calls are the most efficient way to communicate	
Provides context to issue resolution	

COMMUNITY SUPPORT

OVERVIEW

Nothing retains users like an engaged app community. Nurturing this community can be the difference between life and death for an app and some apps can't function without an interactive community. **User to User Chat, Geo Tagging, Screen Sharing** and **VoIP** are vital to community interaction. Whether it is one user posting a question or a group discussion, extensive user to user interaction fosters app loyalty and nurtures a community of users.

Privacy is a key consideration for community support / interaction. For example, let's look at an education app. For any particular student, there will be a set of classes with a set of teachers, a grade level, and a school. It is important for these types of apps that user interaction be limited to the community that is relevant for that user.

KEY FEATURES

1) App to app communication interface, VoIP or chat based

2) New message notifications

3) Ability to follow / unfollow community chats

4) Ability to flag questionable content

5) Agent moderation portal

Community support solutions utilize tagging to limit the availability of the questions to only users that meet the tag criteria. In other words, for our example, only students and teachers that share the tag Broadview High School, Grade 11, Honors Spanish, would be able to ask and answer questions of each other. App based tagging is key to the proper routing of community interactions.

Anytime app users interact with each other and create content, there is a risk for abuse. To manage this risk, the app publisher needs to maintain a moderator.

Key features for app moderation are:

1) Ability for users to flag messages or parts of messages
2) Automatically hide flagged messages
3) Ability for moderator to review and accept flagged messages

PROS AND CONS OF COMMUNITY SUPPORT

+	-
Engages customer	Time consuming for user
Built into app experience	Requires moderation
Scales well - one person can handle significant community moderation load	

OTHER IMPORTANT CAPABILITIES

There are several other areas, outside of the core support stack feature set, that are important to the overall operation of your in-app support solution.

PLAYS WELL WITH EXISTING ENTERPRISE SYSTEMS

An in-app support solution is just one part of most companies overall support strategy. For some app publishers, this system can be the system of record, but for many, it must play well with an existing ticketing system utilized by the company's other products.

Best practice integretion usually supports the following workflow.

1) A question initiated in the app creates a corresponding ticket in a third party help desk. The best practice would be for the ticket to get created after an automated system has attempted to answer it.
2) The third party help desk provides agent assignment.
3) The third party help desk will provide a ticket screen for the agents that should provide a link to the in-app support tool for that ticket.
4) When the agent clicks the link they are taken to the in-app support workbench where they can chat, talk, and troubleshoot the customer's issue. Context must be maintained throughout the entire process.
5) Ticket state and ability to close the ticket should be available in both locations.

Common help desk solutions include ones from Salesforce.com, SAP, Oracle, Zendesk, Freshdesk, and others.

SUPPORTS FOR THE TECHNOLOGY YOU DEVELOP WITH

There are many types of technology platforms used to develop apps. The most common are the platforms for the phone itself, Xcode and Swift, for the iPhone, Java for Android. In addition to these platforms, there are hybrid platforms that allow a developer to create an app in their environment and target it to either or both of the popular platforms. Examples of these development platforms include Xamarin, Cordova, and Unity.

ANALYTICS

In-App support is a rich source of information concerning the use of your application. Through this lens, you can know a complete heat map of issues: which ones are most common, what types of users do they impact, and where in the app do they occur. You will know what issues can be handled automatically and which ones require escalation. This information is very valuable to app design personnel, help desk operations, and is critical to the overall performance of the app.

SECURE

In-App support provides a link between your customers and your organization, so it warrants careful consideration from a security perspective. Fortunately, in-app architectures are very secure by their very nature with their interactions limited to the app's sandbox.

Other important considerations are:

1) **Encrypt Everything**

   Today this is pretty much a no-brainer, but we are often surprised by the number of companies that still neglect it. All communication should be encrypted and be done with known end-points.

2) **Customer Permission**

   We advocate permission based interaction where the communication from the server or community is only available when a ticket is open. The presence of a ticket then provides the first layer of permission. For chats, this is all that is required, but for session based interactions, a second level is desired where the user specifically accepts the session (much like answering a call).

3) **Keep Email and Phone Numers Private**

   One of the biggest benefits of in-app support platforms is that they do not require customer private data to communicate. The app and customer request itself provides the necessary information required to open a communication channel. Phone numbers and emails are not required, and as such, cannot be leaked.

WHAT TO THINK ABOUT

When selecting an in-app support platform for your organization it is important to recognize the long-term nature of the selection. Your vendor should have a complete support vision and provide you with a path to implement all the aspects of a complete solution. Most organizations take a crawl, walk, run approach for their support implementations, but you don't want to find yourself in a position where you are unable to meet your business objectives because your provider has painted you into a corner.

We find the following questions to be useful during the in-app support evaluation process.

1) Does my provider have a full stack solution including no touch, low touch, and high touch solutions?
2) Will I be able to solve problems as well as capitalize on opportunities utilizing this platform?
3) Is the solution easy to integrate with my app?
4) Can I customize workflow and screens to make sense for my app?
5) Can I integrate the platform with my current helpdesk?
6) Is the platform secure?