Top 10 Reasons Why Developer Programs Fail

A white paper by:
Brian Lawley
President, 280 Group LLC

The Product Marketing and Product Management Experts™

About the 280 Group…

The 280 Group LLC provides consulting, contracting, training and templates to help companies define, launch and market breakthrough new products. For more information or a free consultation call 408-832-1119 or visit their website at www.280group.com.

To receive future white papers like this one or to subscribe to the free 280 Group newsletter visit www.280group.com.

Copyright 2006 280 Group LLC. All rights reserved, including the right of reproduction in whole or in part of any form.

Table of Contents

About This White Paper

This white paper is part of the 280 Group Developer Program Toolkit™, which provides you with the training, knowledge and templates to help you create and run a successful developer program. The content of this white paper is at a fairly basic level - the toolkit includes much more in-depth information.

The Developer Program Toolkit includes:

- A 49 slide seminar with narration
- Guideline Documents:
    - Developer Evangelism Plan Guidelines
    - Developer Briefing Guidelines
    - 3 Year Roadmap Guidelines
    - 1 Year Roadmap Guidelines
- Templates & Samples
    - Evangelism Plan Template
    - Sample Evangelism Plan
    - Seeding Program Invitation Letter
    - Developer Database (Excel & FileMaker formats)
    - Evangelism Project Cost Estimator
    - Evangelism Timeline
    - Developer Briefing Agenda
    - 3 Year Roadmap
    - 1 Year Technology and Tools Roadmap
    - 1 Year Developer Training Roadmap
    - 1 Year Marketing Opportunity Roadmap
    - Partner Program Agreement
- A 30 minute consultation to discuss your Developer Program plans with the 280 Group experts.

For more information on the Developer Program Toolkit go to www.280group.com.

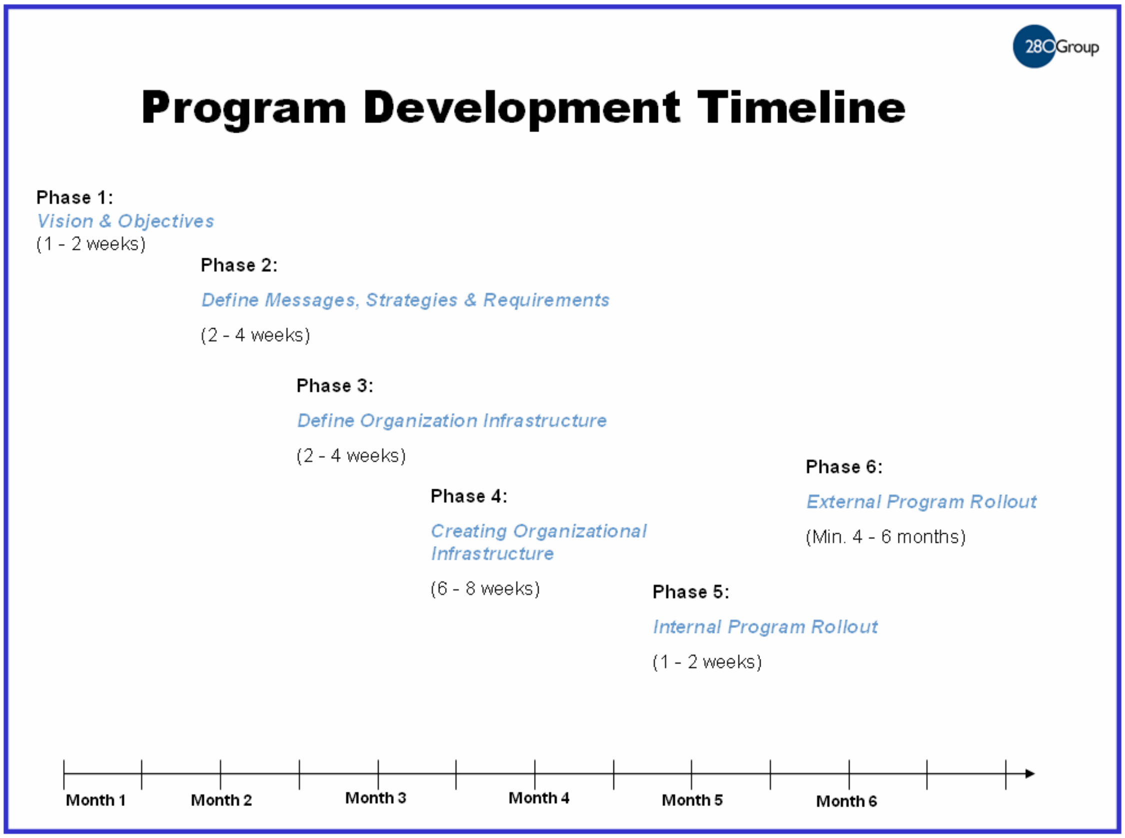
Introduction

An effective developer program can help make a company very successful. Through great developer programs companies like Microsoft, Apple, Palm, Sun (Java) and others have been able to leverage the power of developers and reap great profits and strong strategic positions in the industry. Yet most companies that try to create and run developer programs fail for a variety of reasons.

At the 280 Group we've been involved with creating and executing developer programs for many companies, and we've learned over the years what it takes to be successful. This white paper will outline the top 10 most common mistakes that companies make when running developer programs, and will give you some guidelines and advice based on years of experience so that you don't make the same mistakes.

1. Underestimating the time required

Many companies drastically underestimate the time that is required to create and launch an effective developer program. Realistically it will take anywhere from 4 to 6 months to get a program off the ground, and you should count on it taking months (and probably years) before you attract a critical mass of developers to your platform. The diagram below gives an approximate idea of what to expect in terms of timing.



Program Development Timeline

Phase 1:
Vision & Objectives
(1 - 2 weeks)

Phase 2:
Define Messages, Strategies & Requirements
(2 - 4 weeks)

Phase 3:
Define Organization Infrastructure
(2 - 4 weeks)

Phase 4:
Creating Organizational Infrastructure
(6 - 8 weeks)

Phase 5:
Internal Program Rollout
(1 - 2 weeks)

Phase 6:
External Program Rollout
(Min. 4 - 6 months)

©2006 280 Group LLC. All rights reserved.

**An effective Developer Program will take
4-6 months to plan and launch**

2. Not bringing developers in early enough

If you are developing APIs and a platform it is critical that you get feedback as early as possible (ideally during the architecture design phase). If you develop the software in a vacuum without getting solid feedback you risk creating something that developers won't want to adopt. Showing them what you are up to early will help you adjust your course if need be and make changes that give you a higher probability of success. The other benefit of bringing them in early is that they will feel some ownership over the results, and will be more likely to be evangelists for your platform.

3. No compelling value proposition

Oftentimes companies will go to developers with "Cool" technology, but won't have a compelling value proposition. No matter how great your technology is, you won't get large numbers of developers to take advantage of it unless you can clearly show them how they can make a lot of money. Just because it is interesting, cutting-edge technology doesn't mean you'll be able to win their mindshare (or their time) to create solutions with it.

4. Underestimating the money and resources required

Getting a developer program up and running is going to take a massive amount of effort. Not only will you need dedicated people to design and run the program, you'll need evangelists to recruit developers, first-rate support staff to help them with questions and a good deal of time from all of the other parties who have a stake (Product Management, Marketing, PR, etc.). Many companies think they can assign one person to the task of creating and executing a program, only to find out after the fact that they have woefully underestimated. Developer programs also require significant funding - plan on spending tens (or more likely hundreds) of thousands of dollars on marketing programs, incentives for developers and other expenses related to getting the program momentum going.

5. Focusing on ROI rather than strategic value

Is your developer program a short-term marketing or revenue generating initiative, or is it a longer-term strategic initiative? The fact is that it is very difficult to quantify the ROI for a developer program in the short-run. If you think of it this way you will inevitably give up on the program before it has the chance to make a big impact on your efforts. Recruiting developers is a long-term play that can give you a strong strategic advantage, so don't make the mistake of expecting it to pay for itself in just one or two years.

6. Not having a well-thought out plan

This would seem to be obvious, but you would be surprised at how many companies we come across that are trying to get a developer program off the ground without having really thought it through. This approach is worse than doing nothing at all. Not only are you likely to fail, but you may alienate developers and never be able to bring them back into the fold at a later date. Once your company loses credibility with developers it is virtually impossible to regain it - they are a tough crowd and will not be forgiving if you break commitments.

7. Not having answers to the tough questions

One sure way to fail is to launch the program and begin attempting to brief and sign up developers without having answers to the tough questions beforehand. If you can't answer the following questions you aren't ready to roll out the program:

- Why should I develop for your platform?
- How are you going to help me be successful?
- How much will it cost me to participate?
- How do I get started?

8. Not having adequate support in place

Don't launch your program until you have first-rate technical support people in place who are familiar with the platform. Nothing will piss off a developer more than getting stuck, trying to get help and getting an incorrect or partial answer. Make sure that your technical documentation is excellent, that you have FAQs and a knowledge base in place and that your engineers are committed to helping the support staff make developers successful.

9. Making the program free

If you allow all developers into your program for free you'll end up with a large number of semi-committed partners. It is much better to charge a fee for your program in order to weed out the developers who would be wasting your time. The fee doesn't have to be a lot, and you can waive it for the first N developers and/or offer a free membership to select developers. But make sure you assign a value to it so that you know that those who choose to participate are serious.

10. Providing inadequate tools

Nothing will stop your program in its tracks faster than failing to provide great tools. If your tools aren't ready when you want to launch, then delay the launch. It is better to come out slightly later with tools that make your developer base highly productive than to provide them with half-baked or unfinished tools that they get frustrated with. This ties back into building credibility - if you give them solid tools and excellent support they are much more likely to stick with your platform and build great applications to support your efforts.

Next Steps

This white paper has discussed some of the biggest mistakes that companies make in creating and running developer programs. For in-depth additional developer program training, tips, best practices and templates be sure to visit www.280group.com to take a look at the 280 Group Developer Program Toolkit. And if you need assistance with developing or running your developer programs or need an external opinion on your plans contact the 280 Group and we can provide you with a seasoned professional to help you with your efforts.