

# Platform Management Tool

## Dimension One: Developer Canvas



**Developer: Develops extensions, modules such as apps using the platform. Can be external or internal to the firm.**

### Focus Areas

### Entry Barriers; Ability to Innovate; Boundary Resources; Openness; Feedback

<b>Entry Barriers</b> 	<b>Technology</b> 	<ol style="list-style-type: none"> <li>1. Openness</li> <li>2. Accessibility</li> <li>3. Programming languages</li> <li>4. Toolkit</li> <li>5. Developer type</li> <li>6. Stickiness</li> <li>7. Homing costs</li> <li>8. Documentation</li> <li>9. Standards and protocols</li> <li>10. Support</li> <li>11. Usability</li> <li>12. Developer satisfaction</li> <li>13. Developer context</li> </ol>	<ol style="list-style-type: none"> <li>1. Will the level of architectural openness limit the developers to develop their own products and services? Is the level of openness a potential threat to the platform architecture in terms of duplication or theft?</li> <li>2. How will developers access the platform?</li> <li>3. Will the programming language required to develop products and services be difficult to learn? In the case of a non-popular programming language, why will the platform benefit by using it?</li> <li>4. What features should be included in a toolkit to ensure quality and ease of use of the platform? From a developer's perspective, what are the requirements of a toolkit to enable complementary products and services to be developed on the platform?</li> <li>5. What would a typical developer aim to achieve on the platform? What type of app would developers develop (enterprise, commercial, personal, web-based, hybrid, native apps)? Further extensions, modules, added functionality?</li> <li>6. How difficult will it be for a developer to leave the platform (lock-in)? Will this have an effect on a developer's motivation to join this platform?</li> <li>7. What are the complete set of costs associated with joining the platform? Will access to the platform and its interfaces be restricted based on costs (different packages)?</li> <li>8. Are there sufficient documentation available to describe the comprehensive use and functionalities of the platform?</li> <li>9. Does the platform use and follow well-known and industry accepted standards and protocols?</li> <li>10. Does the platform provide internal and external support? Internal refers to platform owner firm's dedicated support team and external refers to communities outside of the platform firm</li> <li>11. How usable is the platform from a developer's perspective? Usability can refer to learnability, understandability, etc.</li> <li>12. Can the satisfaction of developers be measured? Is it enjoyable to develop complementary products and services on the platform?</li> <li>13. What is the context of use of the envisaged developers? Do they have access to the latest technologies, laptops, connectivity or resources?</li> </ol>
	<b>Mission</b> 	<ol style="list-style-type: none"> <li>1. Trust</li> <li>2. Reputation</li> <li>3. Credibility</li> <li>4. Loyalty</li> <li>5. Fairness</li> </ol>	<ol style="list-style-type: none"> <li>1. How can trust be fostered within the ecosystem? Is the platform conveying a sense of trust (not constantly enveloping)? Is there a sense of trust between (1) platform owner firm and developers, (2) platform owner firm and end users, (3) developers and end users?</li> <li>2. How do external parties perceive the platform, the platform ecosystem and the platform owner firm? Is this a desirable view?</li> <li>3. Do the platform and platform owner firm do what they set out to do? Is the platform brand and reputation credible?</li> <li>4. How can the platform owner firm invest to maintain the loyalty of developers? What are rival ecosystems doing to attract developers?</li> <li>5. Are the dealings within the ecosystem fair (referring to monetary and proprietary aspects)?</li> </ol>
	<b>Value Configuration</b> 	<ol style="list-style-type: none"> <li>1. Value creation</li> <li>2. Value distribution</li> <li>3. Pricing strategy</li> </ol>	<ol style="list-style-type: none"> <li>1. How will developers create and share value within the platform ecosystem? What are ways in which value can be co-created? Can the developers create adequate value to motivate joining the ecosystem?</li> <li>2. How will value (profits) be divided and distributed throughout the ecosystem?</li> <li>3. What is the revenue model and will it be beneficial for complementors? Such as pay fixed amount, percentage, subscription, licencing fees, etc.</li> </ol>
	<b>Ecosystem</b> 	<ol style="list-style-type: none"> <li>1. Market size</li> <li>2. Markets</li> <li>3. Envelopment</li> <li>4. Diversity</li> <li>5. Industry specific resistance</li> </ol>	<ol style="list-style-type: none"> <li>1. What are the global footprints of the platform and its ecosystem? What is the potential market that a developer will be able to access when joining the ecosystem?</li> <li>2. Will the end-products or services be distributed via an application portal/marketplace? Will developers be able to distribute freely on these portals if using the platform?</li> <li>3. How and when will new functionalities be incorporated into the platform? How will developers be encouraged to join the ecosystem if the platform regularly adds functionalities which might envelope a developer?</li> <li>4. Are there a healthy variety of developers using the platform? Will this encourage developers to join and share value, or result in unhealthy competition within the platform ecosystem?</li> <li>5. Are there industry-specific barriers to adoption? For example medical doctors concerned about legal implications when joining a tele-health platform</li> </ol>
<b>Ecosystem</b> 	<ol style="list-style-type: none"> <li>1. Tensions</li> <li>2. Partner interests</li> <li>3. Network effects</li> <li>4. Encourage innovation</li> <li>5. Co-evolution</li> <li>6. Attraction</li> </ol>	<ol style="list-style-type: none"> <li>1. Could risks taken by the platform owner firm (regarding evolution, financial, scope boundaries) affect the complete ecosystem and thereby result in tensions between ecosystem participants? Relate to risk management and how it is shared by all parties in ecosystem</li> <li>2. Does the platform owner firm take the best interests of the both themselves and developers into account when making decisions? Does the platform owner firm act in a selfish way at the cost of other ecosystem actors?</li> <li>3. How will network effects be encouraged amongst developers? Are there any sign of negative network effects within the ecosystem? How do the ecosystem size and scope affect the prevalence of network effects?</li> <li>4. How can innovation be shared with developers? How can innovation from developers be encouraged? Will external innovation lead to less risk and improved R&amp;D?</li> <li>5. Is the platform co-evolving with the complete ecosystem? Should ecosystem goals, vision and current markets be re-evaluated?</li> <li>6. How will planned and potential groups of actors be attracted to join the ecosystem?</li> </ol>	
<b>Technology Infrastructure</b> 	<ol style="list-style-type: none"> <li>1. Interfacing/compatibility</li> <li>2. Feedback</li> <li>3. HW and SW integration</li> <li>4. Marketplace requirements</li> <li>5. Leveraging</li> <li>6. Developer practice</li> <li>7. Vulnerability</li> </ol>	<ol style="list-style-type: none"> <li>1. To what extent will compatibility with other platforms or systems be enabled or allowed? What level of control will be maintained with regard to interfacing of developer products and services with other systems, solutions or components?</li> <li>2. Is there a method of obtaining feedback from developers regarding the platform architecture, interfaces, support, etc.? Will developers be encouraged to provide feedback regarding the platform?</li> <li>3. Do the hardware devices on which the platform and its end-products will be used have an effect on the platform design? Should the platform be able to integrate with existing software systems or components in order to fulfil its function? Should specific data formats or protocols be accommodated?</li> <li>4. How can the platform owner firm design its platform to enable developers to deliver better apps at their preferred marketplaces? What are the specific marketplace requirements?</li> <li>5. How can the developers be leveraged to enable scalability and adaptability of the platform and the ecosystem? Are there developer functionalities that can be used by the platform?</li> <li>6. Is developer training required to use the platform? How will developer best practices be encouraged and enforced if necessary?</li> <li>7. What could be possible weak points in the platform software that could endanger the platform? Could developer laziness compromise platform security?</li> </ol>	
<b>Control</b> 	<b>Rules and Regulations</b> 	<ol style="list-style-type: none"> <li>1. Policies</li> <li>2. Intellectual Property</li> <li>3. Data privacy and security</li> <li>4. Data governance</li> </ol>	<ol style="list-style-type: none"> <li>1. Are there any current governmental or organisational policies that the platform owner firm and developers should be aware of?</li> <li>2. How can the platform owner firm help developers protect their own IP?</li> <li>3. Should the personal data of developers be protected from the platform owner firm and from other developers? Can developers be assured of the safety of their data?</li> <li>4. Who owns the data generated by the developers' products and services? Could data ownership lead to information asymmetry?</li> </ol>
	<b>Performance (Formal and Informal)</b> 	<ol style="list-style-type: none"> <li>1. Control mechanisms</li> <li>2. Design rules</li> <li>3. Goal congruency</li> <li>4. Monitoring and evaluation</li> <li>5. Track user loyalty</li> <li>6. Review process/content regulation</li> </ol>	<ol style="list-style-type: none"> <li>1. What control mechanisms could be implemented to encourage desirable behaviours in developers? Content regulation, review processes, documentation, design guidelines etc.</li> <li>2. What are the fundamental rules that developers should obey when using the platform? Are these design rules stable and consistent? Do they avoid favouritism amongst developers? Are these design rules versatile enough to allow for developer innovation?</li> <li>3. Are the goals of developers and the platform compatible (to reduce the possibility of future tensions and competition)? Could aligned goals lead to increased productivity and innovation?</li> <li>4. How can the ecosystem and its participants be monitored and their performance evaluated? How can M&amp;E be used in managerial decision-making processes?</li> <li>5. Is the number of active developers increasing or decreasing? Is there an external competing ecosystem causing users and developers to migrate?</li> <li>6. Will the complementary products/services be reviewed by the platform owner firm as a method of quality control? Who will be reviewing these components? Are the developers aware of this reviewing process? What are the conditions to be met to "pass the test"?</li> </ol>
<b>Support</b> 	<b>Community Support</b> 	<ol style="list-style-type: none"> <li>1. Online communities</li> <li>2. Ability to share and innovate</li> </ol>	<ol style="list-style-type: none"> <li>1. How can connectedness and a sense of community between developers in the ecosystem be encouraged and facilitated? Can developers learn from each other and build on top of each other's products and services?</li> <li>2. Is there a method of sharing knowledge within the ecosystem? How can innovation be encouraged amongst developers?</li> </ol>
	<b>Platform Support</b> 	<ol style="list-style-type: none"> <li>1. Migration convenience</li> <li>2. Internal customer support</li> <li>3. Design guidelines</li> <li>4. Debugging aids</li> <li>5. Testing support</li> </ol>	<ol style="list-style-type: none"> <li>1. How can the migration convenience from a competing ecosystem be supported? How can the platform owner firm add to the convenience when joining from a different ecosystem?</li> <li>2. Is there a dedicated team within the platform owner firm to supply the required support for users of the platform?</li> <li>3. Can the platform firm and developers learn from major development tools and user interface guidelines (such as Apple HIG and Android UI guidelines)? Are there existing software frameworks that developers can use? What existing platforms can the developers interface with?</li> <li>4. Will debugging aids be developed and be a part of the toolkit provided to developers?</li> <li>5. Which forms of testing support could be supplied to developers?</li> </ol>