Successful initialisation

- **Leadership Skills and Behaviours**
  - **Assertiveness**
    - *Rationale* - To ensure **code quality**, **announced milestones**, **community culture** and other critical elements of an Open Source Project (OSP) a **strong but thoughtful leadership** is necessary
    - *Challenge* - Since OSPs are based on the **voluntary work** of contributors and thus their **personal motivation** it is a delicate task to **direct people and enforce decisions**

Leadership Skills and Behaviours

- **Commitment**
  - *Rationale* - Every leader of an OSP must show an **above-average dedication** to work on the project **investing time** for the further development of the software and the community
  - *Challenge* - The huge engagement of the OSP leaders **requires a lot of time** for which **they usually aren't paid** because it's non-productive work in a monetary sense
- **Communicativeness**
  - *Rationale* - To spread the large amount of knowledge OSP leaders need to **communicate well** in order to **motivate potential contributors** to get into the project
  - *Challenge* - **Serious written communication is very laborious** in general and specifically in OSPs **not the main skill** of many software developers

Leadership Skills and Behaviours

- **Experience**
  - *Rationale* - In order to give an **introduction for beginners** and also to **help in complex problems** OSP leaders **need a vast experience with the source code** of the project
  - *Challenge* - Since OSPs usually consist of **thousands and more lines of code** programmers must be part of the developing community for a **long time** to gain sufficient experience
- **Helpfulness**
  - *Rationale* - To **attract newcomers** and **facilitate their entrance** into the project **helpful leaders** are necessary since often **there isn't enough documentation** available yet
  - *Challenge* - **Time of skilled OSP leaders is limited** so **interested persons must show sufficient personal effort before they can expect support from experienced community members**

Leadership Skills and Behaviours

- **Openness**
  - *Rationale* - For OSP leaders openness is necessary in various forms like being **open for beginners and new ideas** and to communicate openly as far as possible
  - *Challenge* - Openness **is not a basic human mentality** so for example to **accept solutions of others** or to be open to leave a project requires a **high degree of maturity** of the leader
- **Patience**
  - *Rationale* - **The growth of a healthy community takes long** since experienced contributors have to introduce newcomers and show them the software and the community culture
  - *Challenge* - Eager developers not only need to endure the low growth rate but also should **be patient enough to e.g. answer similar beginner questions several times**

Leadership Skills and Behaviours

- **Personality**
  - *Rationale* - The **charisma** of OSP leaders **helps in communication** and by **fascinating potential contributors** for their project and thus **increases the attraction of being part of the community**
  - *Challenge* - Skilled developers **may not have a naturally charming character** so they need to substitute it with other specific traits of personality while still remaining themselves
- **Presence**
  - *Rationale* - A **constant presence in the chat room, mailing list or other communication channel motivates new contributors to join the project** and not feeling left alone
  - *Challenge* - It requires conviction for the project and a **longlasting endurance of the leaders to stay in the OSP over several years** and keep on going to be active in the community

Leadership Skills and Behaviours

- **Programming**
  - *Rationale* - **High programming skills are required of all OSP leaders** since e.g. **to be able to help out in architectural issues** an in-depth understanding of the software is necessary
  - *Challenge* - Besides a certain natural programming talent also a **high education and long time experience in software development is required** to fulfil this leadership skill
- **Responsibility**
  - *Rationale* - Compared to developers interested in a specific topic the OSP **leaders bear the overall responsibility for the success of the project** including support of new contributors
  - *Challenge* - There are various demanding aspects such as **answering leftover emails** or being responsible for a **clean and friendly communication and atmosphere**

Leadership Skills and Behaviours

– **Visionary**
  - *Rationale* - OSP leaders need to be **able to communicate a vision for things that are not yet realized** so a clear direction can be fixed as to what and how the project has to grow
  - *Challenge* - Visionary people need to be **creative and generate new ideas** but also **have to be patient** if e.g. the project is not advancing by the expected speed

Successful initialisation

- **Positive Preconditions of the Project**
  - **Programming Language**
    - *Rationale* - The choice of the programming language determines part of the **image**, the **scope of applicability**, the **potential developer community** and other aspects of the OSP
    - *Challenge* - **Since programming languages have huge syntactic and semantic differences a change at a later stage in the OSP is virtually impossible**

Positive Preconditions of the Project

- **Open Source License**
  - *Rationale* - **The type of open source license highly influences the future of the community** because it may promote resp. force or inhibit resp. unbound community building
  - *Challenge* - **For legal reasons and community habits it is unusual to change the license** of the OSP later on so serious thought has to be given to the choice
- **Great Initial Source Code**
  - *Rationale* - The initial source code **determines the attractiveness** of the OSP by **letting potential contributors estimate its potential** thus influencing their decision to join
  - *Challenge* - It is difficult to determine the **best moment when to publish** the project since if it’s too early the potential isn’t revealed and if it’s too late the demand might have gone

Positive Preconditions of the Project

– **Public Demand**

- *Rationale* - The **usefulness of the software for the intended audience** defines its interest in the OSP and thus **determines the motivation to join** the community and start contributing
- *Challenge* - **It's impossible to anticipate the future demand** for an OSP. **The only possibility is to adapt the software to the skill level of the assumed users** to improve the chances of success

– **Degree of Novelty**

- *Rationale* - New OSPs can be **radically innovative**, **partially innovative by introducing new features** or **marginally innovative by improving existing similar applications**
- *Challenge* - One way to increase the novelty of a project is to **implement new standards and technologies** but then there is the **risk of them not becoming mainstream**

Positive Preconditions of the Project

- **Applicability**
  - *Rationale* - The breadth of **applicability determines the potential user community** of the OSP so e.g. **letting the software run on various platforms increases the audience massively**
  - *Challenge* - **Frameworks intend to be broadly applicable** but on the other hand often **require a high amount of initial effort and a lot of programming**
- **Level of Communication**
  - *Rationale* - **In the beginning nothing is known** about the OSP and **the number of involved people is small** so especially in the beginning a **high level of communication is important to attract new contributors**
  - *Challenge* - Usually the OSP **founders prefer to invest time in further development instead of into writing documentation** - also since it becomes obsolete because of software changes

Successful initialisation

- **Promote Community Building**
  - **Modularity**
    - *Recruiting* - **Extensions or plug-ins enable external contributions without much knowledge of the source code** thus attracting different kinds of people and leading to a broad applicability of the OSP
    - *Collaboration* - Because of modularity of the software, **specialization for certain parts of the program is possible among the developers**. Thus **they are able to freely contribute new features to the project whithout dependence on the core developers**
    - *Production* - **Modularization makes large software tightly structured defining clear dependencies among the modules**. When most of the functionality is put into external components **the kernel remains small and robust**. External software can be plugged in and can also be used in other OSPs

Promote Community Building

– **Documentation**
  - *Recruiting* - Documentation fulfils the very **important function of knowledge transfer** thus **enabling newcomers to use the software**. Also **it presents an easy way to start contributing when inexperienced users study the software and write documentation material**
  - *Collaboration* - **Creating qualitatively high documentation is a laborious task** where volunteers are usually rare. Still **it is essential to find people in the community who create complete and well structured documentation**
  - *Production* - To ensure the comprehensibility and thus the longevity of the software **the source code has to be commented completely**. Especially in frameworks **a complete and updated software reference manual is important** to give programmers support to develop their own applications

Promote Community Building

- **Release Management**
  - *Recruiting* - **Frequent releases communicate the progress** of the OSP to the public **raising the attractiveness to participate in the project**. **New features and improvements of the software need to be published visibly for every release**
  - *Collaboration* - New versions of the software have to be released in an appropriate way taking into account the interests of the active and inactive user community
  - *Production* - **The stability and**, if possible, **the backwards compatibility of a new release are very important**. To guard the customizations of user implementations **a clear distinction between external and internal API is essential**

Promote Community Building

- **Collaboration Platform**
  - *Recruiting* - **A high ranking on collaboration platforms** like SourceForge **increases visibility** of the OSP. On the other hand **its own branded website is also important** to give the project a certain individuality
  - *Collaboration* - The responsibility of a collaboration platform is to **provide modern, fast and reliable services for the work** on the OSP. Also **the development team requires a certain freedom to configure the platform for their needs** to efficiently work on the software
  - *Production* - To improve the production level of the OSP **the collaboration platform has to support the development process of the software as much as possible**

Promote Community Building

- **Physical Meetings**
  - *Recruiting* - Presentations of the OSP **show the people behind the project** and also are stimulating moments for visitors to **try the software for the first time**. Personal contacts to the community spreads confidence and motivation
  - *Collaboration* - Personal relationships of contributors are intensified enabling better collaboration. Meetings are also a **good way to accomplish knowledge transfer from experienced members to potential contributors**. In addition, **community meetings serve to take important decisions concerning the OSP and its organisation**
  - *Production* - The idea of a sprint is to **work intensively on the source code and advance the software** by doing bug fixes, architectural improvements and implementation of new features

Promote Community Building

- **Foundation**
  - *Recruiting* - An OSP can benefit from the **good reputation of a well-known foundation** thus increasing the confidence of users. The association also offers a **secure and reliable open source license**, a **correct credit system** and the **protection of the OSPs' brands**. Additionally it may bundle the **marketing power of the community** and **support collaboration among companies** in the OSP
  - *Collaboration* - A foundation has to **bring stability into the project** and to **smooth the fluctuation of people**. It also organizes the various tasks in the OSP, installs **democratic structures** and thus creates **more transparency**. By collecting donations the foundation acquires the ability to **hire people if deemed necessary by the community**. Especially important is the **legal protection of the developers**

Promote Community Building

- *Production* - Besides protecting the software developers against law suits the foundation also has to **secure the copyright of the OSPs' source code**. In addition it may **canalize resources to support the most active core developers so they can concentrate on their work**. The foundation is also responsible to **provide a state-of-the-art collaborative infrastructure** for the OSP

Promote Community Building

- **Internationalisation**
  - *Recruiting* - Going global with the OSP vastly **increases the number of potential contributors**. Additionally it offers an easy **introductory opportunity for newcomers by letting them work on translations**. Also knowing and visiting people from all over the world is attractive for newcomers
  - *Collaboration* - The diversity of the community may help to overcome cultural phenomena like not-so-communicative people in the originating country. Also **knowing of people working on the OSP all over the world at any time of the day has a highly motivational influence on all contributors**
  - *Production* - The software should be **usable in various different languages** and cultural areas

Promote Community Building

- **Spreading the Word**
  - *Recruiting* - General **marketing activities to raise the visibility and improve the image** are also important for OSPs. The goal is to achieve **as much publicity as possible on various open source platforms, directories and news channels**
- **Credit System**
  - *Recruiting* - Instead of paying money OSPs can **offer raising the contributor's reputation**. So **the attractiveness of an OSP raises when participation in the community is acknowledged**
- **Communication Channels**
  - *Collaboration* - For efficient collaboration the optimum mix of communication channels is essential. Therefore, OSP **leaders have to be familiar with strengths and weaknesses of each communication instrument** to deploy the appropriate ones according the community's demands

Promote Community Building

- **Community Structure**
  - *Collaboration* - The investigated OSPs show a broad variety of stages of organisation. Therefore **the appropriate structure has to be found for each project individually corresponding to its size and progress**
- **Task List**
  - *Collaboration* - The idea of a task list in an OSP is to motivate people to contribute to the project **what is actually needed** thus giving them a **hint on how to start participating**
- **Feature and Code Quality**
  - *Production* - Obviously **software quality should be as high as possible**. This holds particularly for open source projects where many different developers have to work with the available code

Promote Community Building

- **DesignUser Interface**
  - *Production* - Although **the actual value of a software is derived from its functionality** the **graphical user interface still plays a major role in the handling and overall look and feel of the application**
- **Installation**
  - *Production* - **The installation process has to be perfectly organized concerning technical installation and documentation of every single step**