

TechLead Project Management with Mobile App Development

Discussion: Project Management & Mobile App Development with Python

Instructor: Xin Wang

Email: xinwang35314@gmail.com

Introducing Project Management

Project :

Converting a vision, a dream or a need to reality.

- ✓ A job that has a beginning and an end (Time)
- ✓ A specified outcome (Scope)
- ✓ At a stated level of Performance (Quality)
- ✓ At a budget (Costs).

Project Characteristics :

- Temporary : Has definite Start and Finish
- Unique : Product/Service is different in some distinguishing way

Management :

Management is the technique of understanding the problems, needs and controlling the use of Resources, Cost, Time, Scope and Quality.

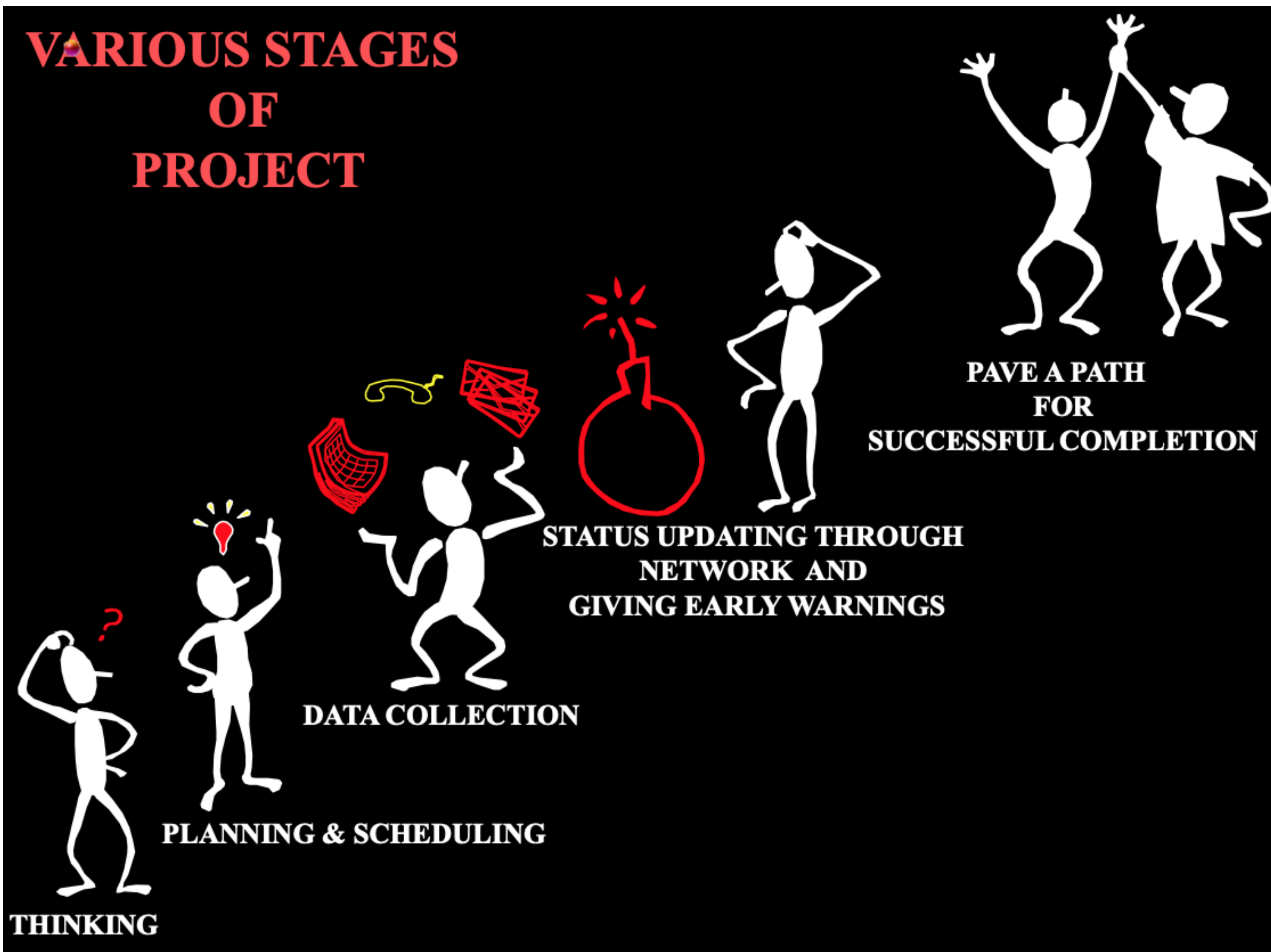
Project Management :

Application of knowledge, skills , tools & techniques to project activities in order to meet stakeholder needs & expectations from a project.

Needs : stated part of the project

Expectations : unstated part of the project

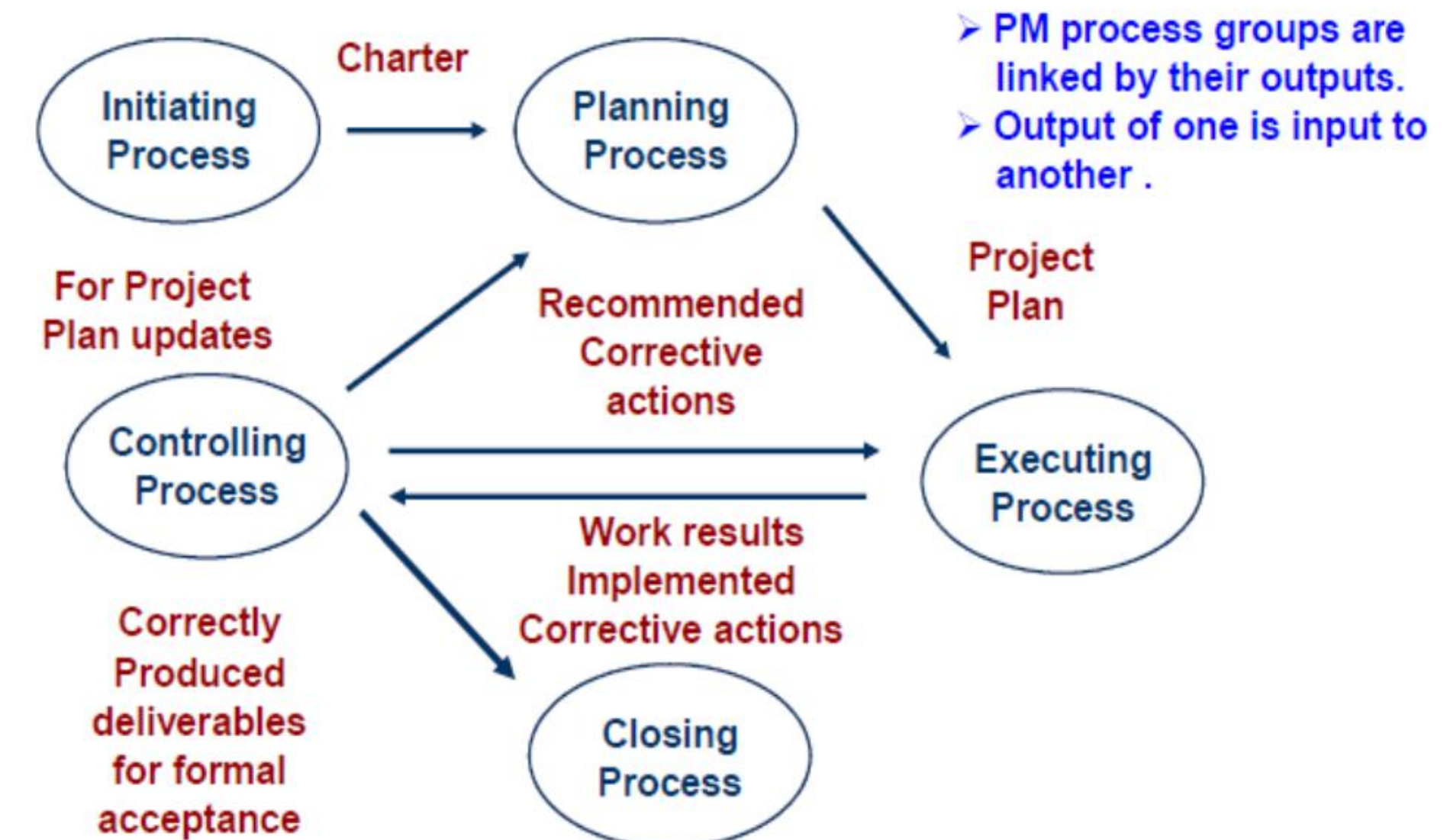
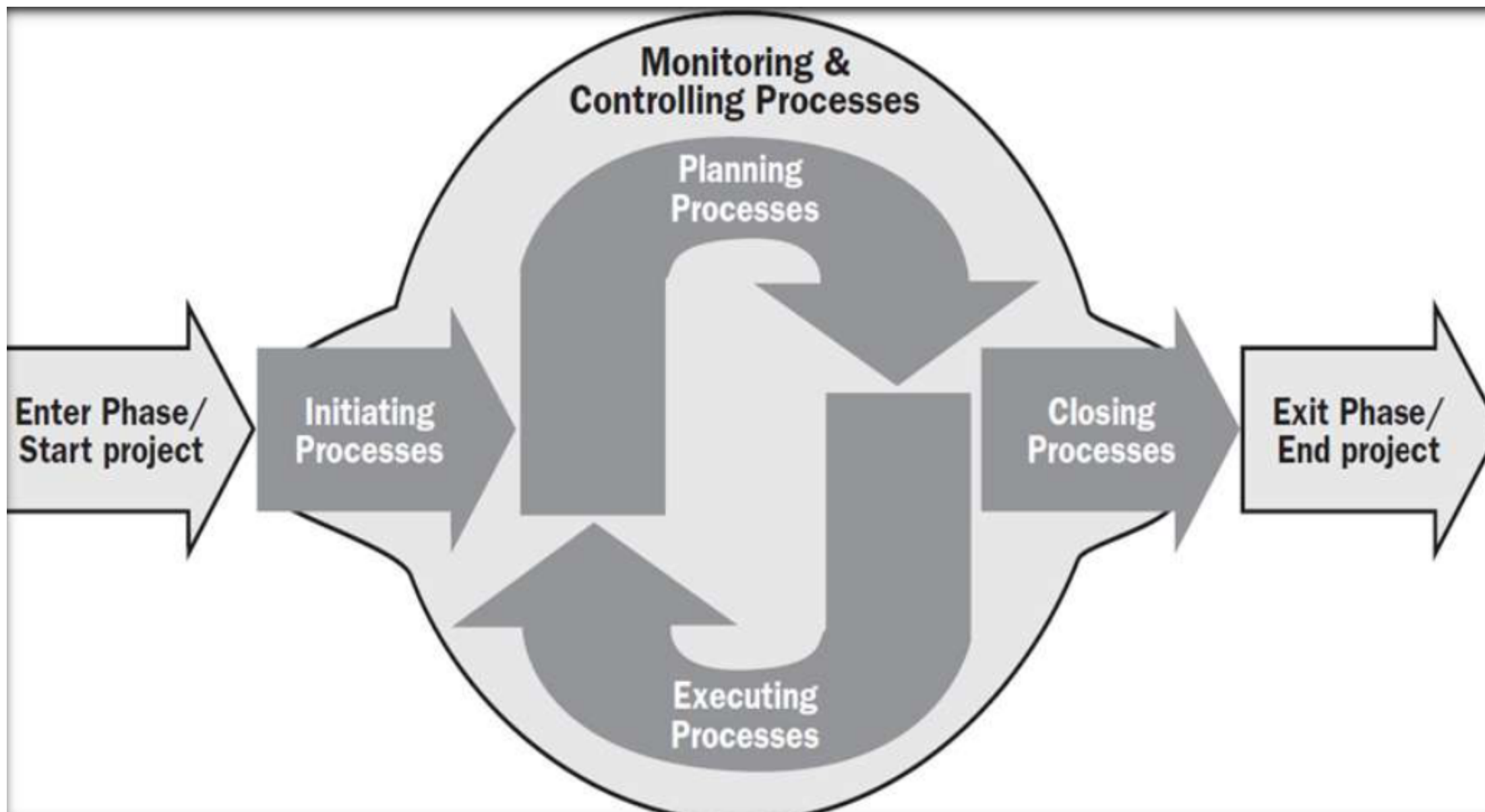
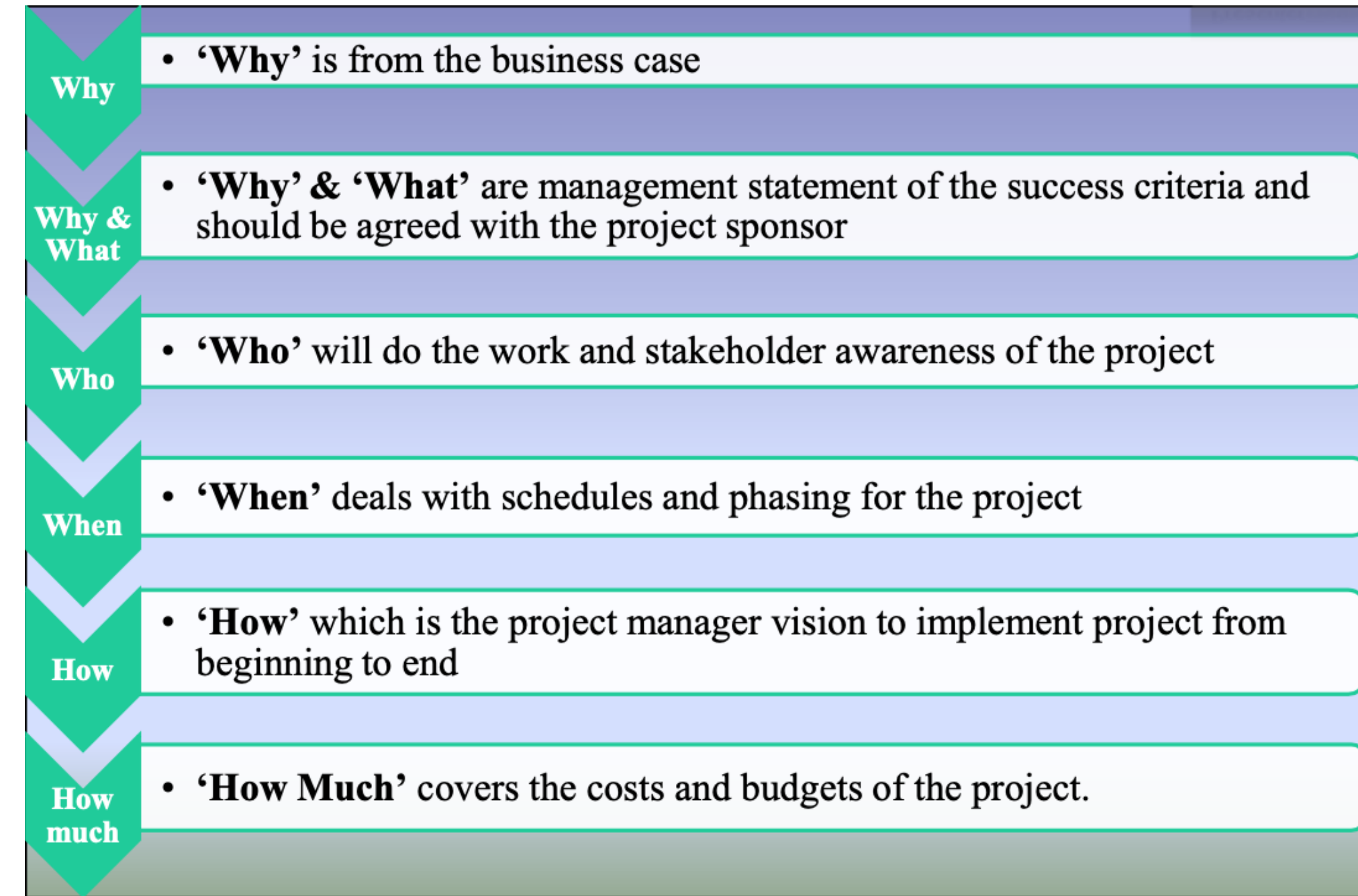
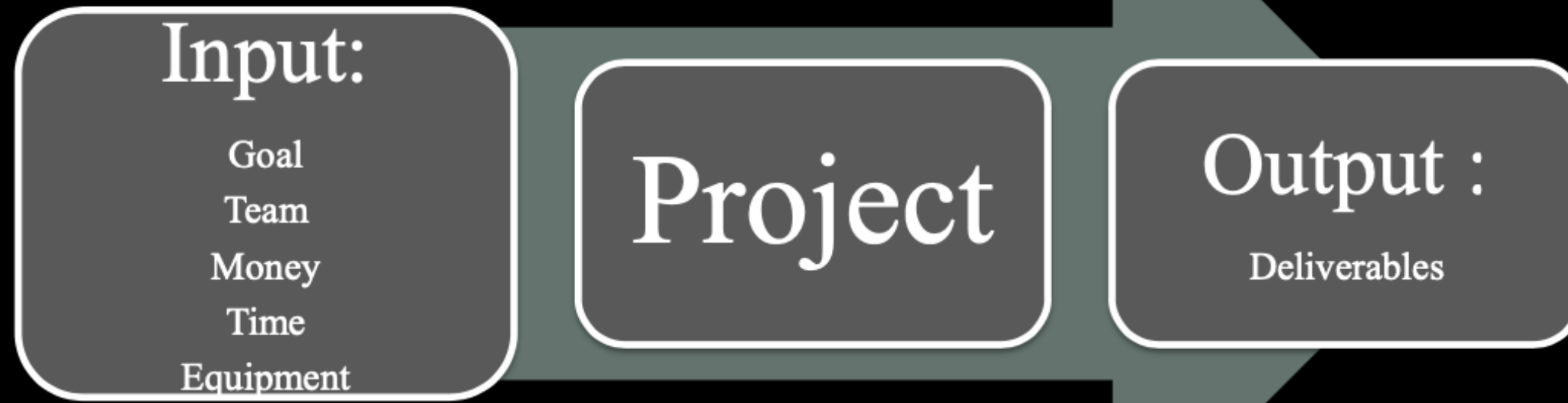
“Completion of Project on time within Budget without comprising **Quality**”



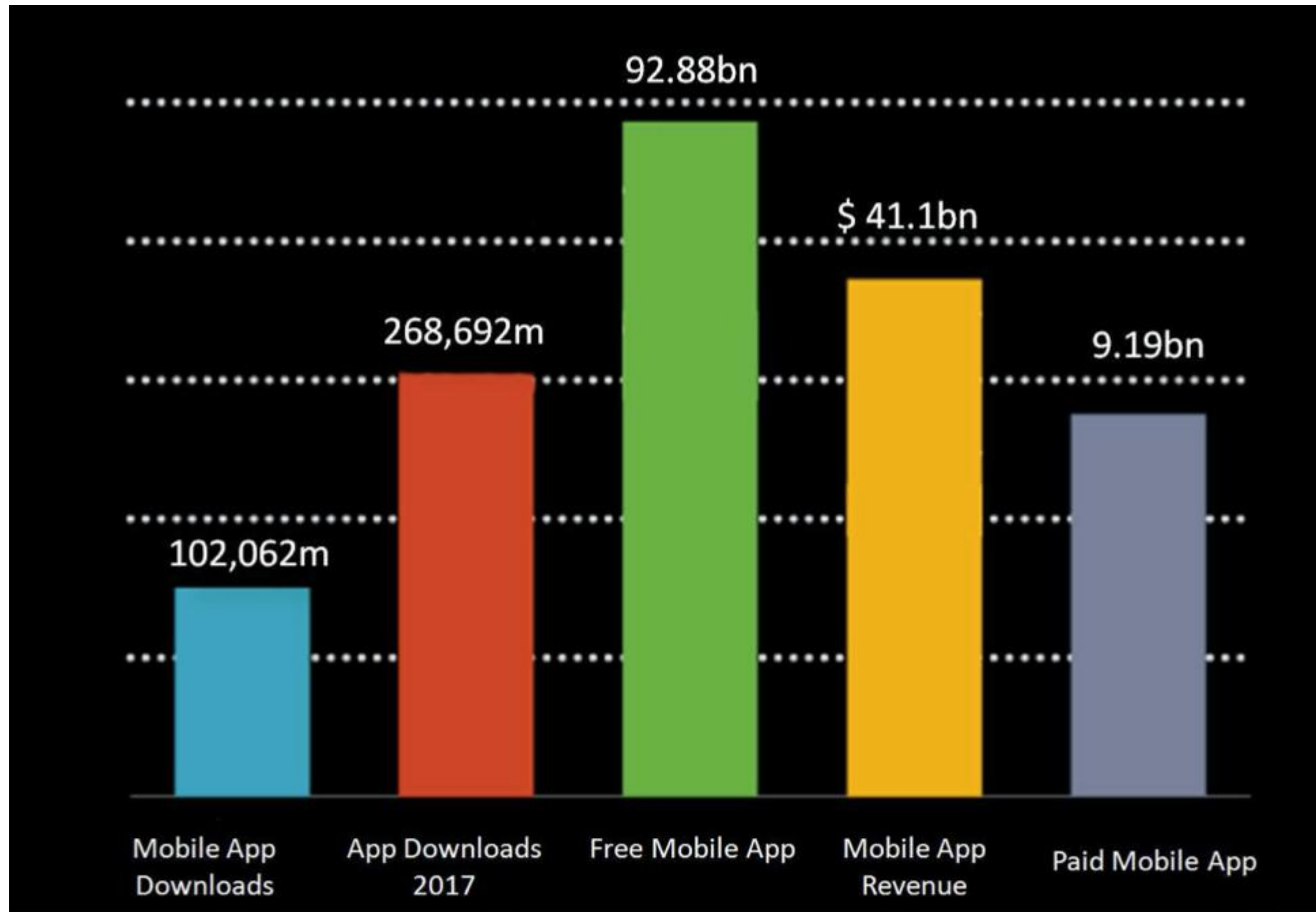
PM Processing

“Tells How work will be done”

The key to a successful project is on the planning. All the detailed planning work for different aspects of the project is integrated into one single plan known as the Project Management Plan.



Why Mobile App?



Do You Know?

“

In 2020

Expected Growth of Mobile App Revenue is **101 Billion** US Dollars.

”

No Matter Large Or Small Enterprise!!

MOBILE APP IS THE NEED OF ALL.....

INITIATION

Initiations talks about the transition of the project from sales team to technical team, where they do initial project kick off with the client and sales team to understand the requirement.

DESIGN

PROCESS FLOW : The Process Flow chart provides a visual representation of the steps in a process. Flow charts are also referred to as ProcessMapping or Flow Diagrams.

WIRE FRAME : Elements visible on each of the screen of the mobile application are discussed and noted down as wireframes.

DESIGN

STORYBOARD : Everything including the images to be shown, icons appearing, theme, color, background, etc is a part of storyboard.

ARCHITECTURE DESIGN : Planning and development of application architecture is performed. Coding is done to give life to the application.

TEST PLANNING : The application is tested for bugs by running various programs, performing checks at different conditions to ensure it is robust and flawless.

SUPPORT

Customer support is a range of customer services to assist customers in making cost effective and correct use of a product. It includes assistance in planning, installation, training, troubleshooting, maintenance, upgrading, and disposal of a product.

TESTING

FINAL BUILD : It's incredibly important to test your application early and often on actual devices. Even devices with the same hardware specs can vary widely in their behavior.

FINAL RELEASE : The thoroughly tested and checked application is finally made available to end users.

Mobile Application Development Process

INITIATION
DESIGN
DEVELOPMENT
ACCEPTANCE
SUPPORT

DEVELOPMENT

PROTOTYPE : The app is still in proof-of-concept phase and only core functionality, or specific parts of the application are working. Major bugs are present.

ALPHA : Core functionality is generally code-complete (built, but not fully tested). Major bugs are still present, outlying functionality may still not be present.

DEVELOPMENT

BETA : Most functionality is now complete and has had at least light testing and bug fixing. Major known issues may still be present.

RELEASE CANDIDATE : All functionality is complete and tested. Barring new bugs, the app is a candidate for release to the wild.

Introducing Mobile App Development with Python – Kivy

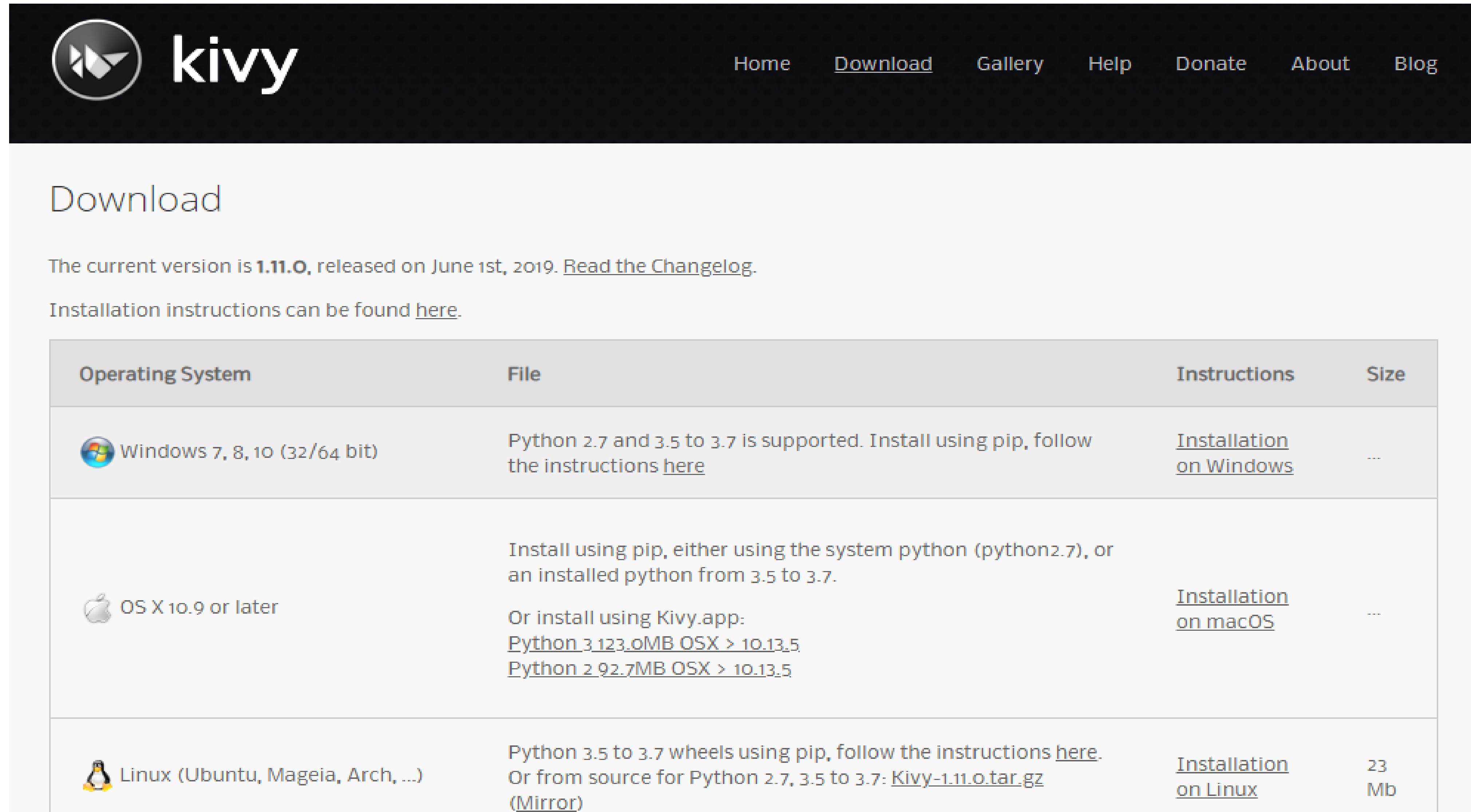
- **Kivy** is an open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps.
- Kivy runs on Linux, Windows OS X, Android, iOS, and Raspberry Pi. You can run the same code on all supported platforms.
- **Kivy language** is a language dedicated to describing user interface and interactions. As with other interface markup language, it is possible to easily create a whole UI and attach interaction.

Eg: create a Loading dialog that includes a file browser, and a Cancel / Load button, one could first create the base widget in Python and then construct the UI in Kv.




```
class LoadDialog(FloatLayout):  
    def load(self, filename): pass  
    def cancel(self): pass
```

Introducing Mobile App Development with Python – Kivy

You can download kivy from <https://kivy.org/#download>



The screenshot shows the Kivy website's download page. At the top, there is a navigation bar with the Kivy logo and links for Home, Download, Gallery, Help, Donate, About, and Blog. The main heading is "Download". Below this, there is a paragraph stating the current version is 1.11.0, released on June 1st, 2019, with a link to the changelog. Another paragraph mentions that installation instructions can be found [here](#). The core of the page is a table with four columns: Operating System, File, Instructions, and Size. The table lists installation instructions for Windows, OS X, and Linux.

Operating System	File	Instructions	Size
 Windows 7, 8, 10 (32/64 bit)	Python 2.7 and 3.5 to 3.7 is supported. Install using pip, follow the instructions here	Installation on Windows	---
 OS X 10.9 or later	Install using pip, either using the system python (python2.7), or an installed python from 3.5 to 3.7. Or install using Kivy.app: Python 3 123.0MB OSX > 10.13.5 Python 2 92.7MB OSX > 10.13.5	Installation on macOS	---
 Linux (Ubuntu, Mageia, Arch, ...)	Python 3.5 to 3.7 wheels using pip, follow the instructions here . Or from source for Python 2.7, 3.5 to 3.7: Kivy-1.11.0.tar.gz (Mirror)	Installation on Linux	23 Mb

Introducing Mobile App Development with Python – Kivy

Kivy Installation:

- Ensure you have the latest pip, wheel, and virtualenv:

```
>python -m pip install --upgrade pip wheel setuptools virtualenv
```

- Create an empty folder for kivy project environment

- Install the dependencies

```
> python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2  
kivy.deps.glew
```

```
> python -m pip install kivy.deps.gstreamer
```

- Install kivy

```
> python -m pip install kivy
```


Introducing Mobile App Development with Python – Kivy

Now let us try our first kivy app!

```
import kivy
kivy.require('1.0.6') # replace with your current kivy version !

from kivy.app import App
from kivy.uix.label import Label

class MyApp(App):

    def build(self):
        return Label(text='Hello world')

if __name__ == '__main__':
    MyApp().run()
```