



0. Introduction to This Syllabus .....	4
0.1 Purpose of this document .....	4
0.2 Cognitive Levels of Knowledge .....	4
0.3 The Examination .....	5
0.4 Business Outcome .....	5
0.5 Specialization .....	5
1. Course Introduction - 15 minutes .....	6
Literature .....	6
2. Introduction to Mobile Application Types and Their High Level Architecture - 60 min (K2) .....	7
2.1 Different types of Mobile Applications – 10 minutes (K1) .....	7
2.2 Mobile Application Architecture – 20 minutes (K2) .....	7
2.2.1 Client-side architecture .....	7
2.2.2 Server-side architecture .....	7
2.2.3 Connection Types .....	8
2.3 Development Environments and Tools – 30 Minutes (K1) .....	8
2.3.1 Mobile Application Development Environment and Tools .....	8
2.3.2 Emulators & Simulators .....	8
3. Introduction to Performance Testing Concepts–70 min (K2) .....	9
3.1 The importance of Performance Testing – 10 minutes (K1) .....	9
3.1.1 Purpose of Performance Testing .....	9
3.1.2 Performance testing focus .....	9
3.2 Key terms and concepts in Performance world – 60 minutes (K2) .....	9
3.2.1 The main concepts used in the Performance world .....	9
3.2.2 The key terms that are used in the Performance world .....	10
3.2.3 The main items required in the Performance test environment .....	10
4. Performance Testing Process, Strategy & Approaches 45 minutes (K2) .....	11
4.1 Challenges in Mobile Application Performance Testing 20 minutes (K2) .....	11
4.2 Performance Testing Process for Mobile Apps 25 minutes (K1) .....	11
4.2.1 Test Process .....	11
4.2.2 Mobile Performance Testing .....	11
5. Performance Testing solutions for different Mobile Applications 605 minutes (K3) .....	13
5.1 Common Performance Issues for Mobile Applications 20 minutes (K1) .....	13
5.1.1 Common Performance Issues .....	13
5.2 Native App Performance Testing 210 (Android)+120 (iOS) minutes (K3) .....	13

5.2.1	Common Performance Issues.....	13
5.2.2	Common Tools and Indicators to be monitored .....	13
5.3	Mobile Web App Performance Testing 130 (Android) + 30 (iOS) minutes (K3).....	14
5.4	Mobile Network Performance 20 minutes (K1).....	14
5.5	Server Side Performance Testing 30 minutes (K3).....	14
5.5.1	Method and Setup .....	14
5.5.2	Monitoring and Analysis .....	15

## 0. Introduction to This Syllabus

### 0.1 Purpose of this document

This syllabus defines the content of the international qualification scheme for the "Certified Mobile App Professional –Performance Testing" (CMAP-PT). It has been established by the Special Interest Group SIG of the International Software Quality Institute (iSQI).

CMAP-PT is an introduction to Mobile application performance testing. It provides an excellent introduction to performance testing in the mobile world for different kinds of apps, the most relevant techniques and terminology.

The iSQI SIG CMAP-PT has created:

- The syllabus
- The Business Outcomes (BO)
- The course material including practical exercises and other artifacts

The course material can be licensed to training providers. In order to license the material the training provider must have at least two trainers that hold the CMAP-PT certificate.

The SIG CMAP-PT-FL qualification is entry level certification aimed at anyone involved in mobile app performance testing: project managers, quality managers, software development managers, business analysts, developers, testers, load & performance testers, IT directors and management consultants.

It is assumed that the trainees have basic knowledge of software testing concepts. It is recommended that the candidate holds a foundation level certificate such as "ISTQB® Certified Tester – Foundation Level" (ISTQB® - CTFL) and Certified Mobile App Professional – Testing" Foundation (CMAP-FL).

### 0.2 Cognitive Levels of Knowledge

Detailed Learning Objectives (LO) are indicated for each section in this syllabus. These objectives identify what the trainee will be able to do following the completion of each module. They are classified as follows:

Level 1: Remember (K1)

Level 2: Understand (K2)

Level 3: Apply (K3)

The top-level heading for each chapter shows the highest level of learning objectives that is covered within the chapter. The definition of these cognitive levels matches the definition given in the ISTQB® Certified Tester scheme to guarantee compliance with and thus integrity to this scheme. Please refer to [CTFL2011] for more details.

## 0.3 The Examination

The CMAP-PT certificate examination will be based on this syllabus. Answers to examination questions may require the use of material based on more than one section of this syllabus. All sections of the syllabus are examinable.

The exam is 60 minutes, 40 questions multiple-choice exam. Examinations may be taken after the training course or taken later (e.g. in a public examination).

## 0.4 Business Outcome

This section lists the Business Outcomes expected of a candidate who has achieved the CMAP-PT Foundation Level certification.

A CMAP-PT-FL professional can

- BO1 Be familiar with the Mobile Application Performance Testing concepts
- BO2 Be able to define the challenges at the Mobile Application domain and in particularly performance
- BO3 Be able to apply the Load & Performance Process in the Mobile Application testing world
- BO4 Be able to define the Performance strategy and approaches when testing different types of Mobile Applications
- BO5 Be able to get familiar with the different attributes of the Performance Testing in Mobile Application
- BO6 Be able to apply Performance Testing for different Mobile Application solutions – Native, Web and Hybrid apps
- BO7 Be able to identify the relevant attributes, how to monitor them and present the results

## 0.5 Specialization

CMAP-PT is one of the family of CMAP certifications that target different proficiency levels as well as specializations. Other certifications from CMAP are listed below:

- Certified Mobile Application Professional – Testing Foundation Level
- Certified Mobile Application Professional – Security Testing
- Certified Mobile Application Professional – Automation Testing

## 1. Course Introduction - 15minutes

### **Literature**

- [KOHL 2013] Tap Into Mobile Application Testing – Jonathan Kohl
- [HART 2010] A Practical Guide to Testing Wireless Smartphone Applications –Julian Harty
- [MILA 2011] Android Application Testing Guide Diego Torres Milano
- [SILLARS 2016] High Performance Android Apps - Doug Sillars
- [VO 2011] Pro iOS Apps Performance Optimization - KhangVo
- [VAISH 2016] High Performance iOS Apps: Optimize Your Code for Better Apps - GauravVaish
- [FIRTMAN 2016] High Performance Mobile Web: Best Practices for Optimizing Mobile Web Apps - MaximilianoFirtman
- [CTFL2011] ISTQB Foundation Level Syllabus
- [CMAP-FLT2012] Certified Mobile Application Professional – Testing (Foundation)

The Certified Mobile Application Professional – Performance Testing(CMAP-PT) certification. Assist in adaptation of current test performance testing processes for mobile app performance testing process. Adapt existing performance testing experience and knowledge to develop performance tests for mobile applications. Identify and Apply appropriate methods for performance testing for mobile applications. Develop and execute performance tests for web, native and hybrid applications using open source tools. Assist in identification of requirements of a test lab for carrying out mobile application performance testing as well as provide instructions and tips for troubleshooting and performance test reports.

The syllabus has following sections –

- Course Introduction (***This section***)
- Introduction to Mobile Application types and their High Level Architecture
- Introduction to Performance Engineering Concepts
- Mobile Application Performance Testing Process, Strategy & Approaches
- Performance Testing solutions for Mobile Applications
  - Performance testing Native Apps – Android and iOS
  - Performance testing Web Apps – Android and iOS
  - Performance testing Web Apps – server side
  - Performance testing hybrid Apps

The exam structure and question distribution is explained as part of the course material. The course timing includes time taken to do the subject discussion as well as the exercises and the exam question distribution follows the timing described in the syllabus.

## 2. Introduction to Mobile Application Types and their High Level Architecture - 60 min (K2)

Terms: Client, Server, Synchronous and Asynchronous connection, Emulators and Simulators, Native Apps, Mobile Web Apps, Hybrid Apps

### 2.1 Different types of Mobile Applications – 10minutes (K1)

#### ***PTFL2.1-1 Be able to recall different types of mobile applications (K1)***

There are various types of mobile applications such as native, browser-based or hybrid mobile applications. Some of the applications come pre-installed on the mobile device and others can be downloaded from respective stores or marketplaces and installed

Each type of application has certain advantages and disadvantages requiring an engineering decision to be made before starting the application development. Testing of each of these application types may require a different approach.

### 2.2 Mobile Application Architecture – 20 minutes (K2)

#### ***PTFL2.2-1 Be able to understand the general architecture (Client & Server) of mobile applications (K2)***

#### ***PTFL2.2-2 Be able to classify the development environment for mobile devices and their tools (K2)***

#### ***PTFL2.2-3 Be able to understand the different connection types and data sync methods that can have an impact on the performance of mobile applications (K2)***

There are multiple solutions to architect a mobile application. Some of the considerations in choosing a particular architecture or design decision are

- Who is the target audience for the application?
- What kind of application we want to build – Native/Hybrid/Web application?
- Is the application meant to run across various mobile and non-mobile platforms?
- What are the connectivity needs for the application?
- What is the data storage need for the application?

#### **2.2.1 Client-side architecture**

Client side application can be Thin-client or Fat-client. Thin client applications do not have customized application code and these make minimal use of the features provided by the mobile operating system whereas Thick/Fat client applications may have multiple layers of application code and may make use of features provided by the mobile operating system. Communication and data storage needs between client and server also plays a role in choosing appropriate architecture.

#### **2.2.2 Server-side architecture**

Server side architecture can be a single-tier or multi-tier. In single-tier architecture all server side components like application server, database server etc. are clubbed into one unit, whereas in multi-tier architecture they are spread across various units.

### 2.2.3 Connection Types

There are various types of connections such as Wi-Fi, Cellular data networks, Bluetooth etc. and data synchronization method such as push and pull. The devices can operate in one of the three modes – Always connected, never connected or partially connected, each mode being useful in certain situations.

## 2.3 Development Environments and Tools – 30 Minutes (K1)

**PTFL2.3-1 Be able to recall the architecture of iOS and Android (K1)**

**PTFL2.3-2 Be able to identify and recall the purpose of some of the common tools that are supplied as part of Android/iOS application development platforms (K1)**

### 2.3.1 Mobile Application Development Environment and Tools

All the operating systems have different set of tools for developing mobile applications. It is useful to know which OS/platform uses which tools and also what host operating system can be used to install and use these tools.

Understanding the platform greatly helps testing of applications on that platform. It is important to get an overview of architecture, storage used, supported programming languages for application development, for major mobile operating systems namely, iOS, Android, Windows Mobile and Blackberry.

There are two major players in the smartphone market currently that provide mobile operating systems. Google, which has Android operating system, and Apple which has iOS. Architecture of each of these OS is layered architecture. The OSes provide various services to application without allowing the applications accessing hardware directly or even through low-level libraries and routines. There are various tools provided with the development platforms to facilitate application development, testing and debugging.

Note - two other popular operating systems by Blackberry Limited (earlier Research in Motion or RIM) that has Blackberry operating system and Microsoft which has Windows Mobile operating system are not covered.

### 2.3.2 Emulators & Simulators

**PTFL 2.3-3 Understand differences between emulators and simulators (K2)** <sup>[1]</sup> <sub>[SEP]</sub>

**PTFL 2.3-4 Understand the application of emulators/simulators for mobile application testing (K2)** <sup>[1]</sup> <sub>[SEP]</sub>

Emulators are very useful in the early stage of development as these typically integrate with development environments and allow quick deployment and testing of applications. Emulators are also used to reduce the cost of test environments by replacing real devices with emulators. However an emulator can't replace a device because the emulator may behave in a different manner than a mobile device. Emulators may not support all mobile device features. In addition some hardware types may not be supported such as touch, accelerometer and others. <sup>[1]</sup> <sub>[SEP]</sub> Simulators too are tools that mimic the device. However unlike emulators, which can consume device executable, simulators require applications to be compiled specifically for these. <sup>[1]</sup> <sub>[SEP]</sub>



## **3. Introduction to Performance Testing Concepts–70 min (K2)**

**Terms: Performance Testing, Load Testing, Stress Testing, Scalability, Use/Load Profile, Throughput, Stability, KPI, Bottleneck, Memory leak, Crash, Deadlock, Latency**

### **3.1 The importance of Performance Testing– 10 minutes (K1)**

*PTFL3.1-1 Be able to recognize the main purpose of the Performance Testing (K1)*

#### **3.1.1 Purpose of Performance Testing**

The main purpose of Performance Testing is to determine how a system performs in terms of responsiveness and stability under a particular workload and mitigate the related risks. The risks can address the ability of the application to handle the expected workload (usually the amount of transactions) which set of users perform concurrently at a given duration; the ability of the application to respond in a required timing; the ability of the application to be available, stable and reliable while load is running (normal and/or stress). Performance Testing can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

#### **3.1.2 Performance testing focus**

Performance testing is focused on generating load on the back end (servers) of the system under test, by using load tools to simulate the transactions run by the client(s) interface(s) that generate the traffic towards the servers using the relevant protocols.

### **3.2 Key terms and concepts in Performance world – 60 minutes (K2)**

*PTFL3.2-1 Be able to explain the various of concepts in Performance (K2)*

*PTFL3.2-2 Be able to categorize the key terms in Performance into business and technical aspects (K2)*

*PTFL3.2-3 Be able to recognize the main items required to be in the performance test environment (K2)*

#### **3.2.1 The main concepts used in the Performance world**

The performance world has its own unique tests and concepts. The performance activities have various concepts that drive them forward. The purpose of the performance testing is to mitigate risks in aspects that related to the way the product – system or application or solution – behave under load (its performance), addressing in addition various non-functional attributes like stability, reliability, availability, resiliency etc. The risks mitigations are measured against the cost of failures, the rate of failure and recovery, response time of the business flows vs. the needed ones in the market. In addition, looking for system bottlenecks and additional failure attributes, such as memory leaks, indication for crashes and deadlocks, limit of module or component's abilities. There are tests need to be performed for the devices and measure the impact on the performance of the application and the device – mainly the defined KPIs for them. As well as for performing test for the Server-side and the impact on the application performance and the server-side performance as a results of these tests – as well against the defined KPIs for them.

### ***3.2.2 The key terms that are used in the Performance world***

The main concepts are to address business needs throughout business processes defined to be measured. These business processes are captured into a load model that includes the usage model (how the users will use it?) and the traffic model (how the components of the system will handle it?); in addition to that, there are figures that placed/targeted for each component in the system in terms of capacity – how much traffic, transactions etc. the component can handle, and by that, we create a map between the transactions and the components and the infrastructure that holds the components. The additional information that is need to be placed before practicing the load model, are the profiles that the system should have, in terms of data, scale of data, complexity of the data, mimicking the real world as much as possible. In a case it's a new product, the testing load manager and the product manager, needs to define the profiles aligned with the business needs/targets that put for this product to succeed. The system shall be load accordingly and then measured against the Key Performance Indicators (KPIs) that defined as success criteria. Each KPI has its own values and trash holds to be measured against. All of the above shall be handled by a process which is specific to the way we handle load and performance tests.

### ***3.2.3 The main items required in the Performance test environment***

There are mainly 3 type of items that required when considering the environment for Performance tests as well as for the production: physical test environment (servers, machines, network devices etc.), its hardware, software and network configuration. Tools (including test tools, monitor tools etc.); and Resources available from the testing team.

## **4. Performance Testing Process, Strategy & Approaches 45minutes(K2)**

Terms: Performance Test Plan, Test Environment

### **4.1 Challenges in Mobile Application Performance Testing 20 minutes (K2)**

#### ***PTFL 4.1-1 Understand and recall the challenges in Mobile Application Performance Testing (K2)***

Performance Testing for Mobile application requires a process to be established like in any other software testing project. One of the main key points is to identify and be aware to the challenges that we've in the mobile application performance-testing domain.

While being a part of these projects you will require to address challenges such as the ability of the application to handle exponential growth. In addition meeting the required performance criteria while addressing various challenges such as limited mobile network bandwidth and other mobile network issues, exponential growth in user sessions, transactions and data transfer; the ability to support multiple devices types, multiple mobile operating systems and platforms, unique user interfaces, various of mobile application types, assessing the capabilities of the client side as well as the server side, the verity of the technology that currently growing such as cloud-based apps, the capabilities of the devices and concurrent apps running simultaneously, the consumption of the mobile's resources while the apps are using different types of sensors and supportive performance tools.

### **4.2 Performance Testing Process for Mobile Apps 25 minutes (K1)**

#### **4.2.1 Test Process**

##### ***PTFL 4.2-1 Recall the phases of the Performance Testing Process (K1)***

The testing process which is applicable to mobile application testing requires the following steps -

- Performance Test Plan
- Identify the Test Environment
- Identify Performance Acceptance Criteria
- Plan and Design Tests
- Configure the Test Environment
- Script Development
- Monitoring Setup
- Execute the Tests
- Result Analysis & Diagnostics
- Report and Retest
- Closure

#### **4.2.2 Mobile Performance Testing**

##### ***PTFL 4.2-2 Recall the Performance Testing Test Objects for Mobile Application Testing (K1)***

##### ***PTFL 4.2-3 Recall the Performance Testing requirements for Mobile Application Testing (K1)***

Mobile App Performance Testing consists of Client side testing done on the device for all types of apps namely, Native Apps, Web Apps and Hybrid Apps and on the server side it consists of usual performance

testing as the desktop/web apps. However, on the mobile devices, the tools used to capture traffic information may be different. Network side performance testing requires the ability to mimic real world usage and conditions of network, signal etc.

Mobile performance testing requires support for device simulation, support for network emulation with an ability to simulate Mobile networks such as Edge, 3G, 4G etc. There might also be a requirement for support of various technologies such as AJAX, HTML 5, JSON, Flex, Oracle Forms, Silverlight, .NET, SOAP, SAP etc.

For the purpose of setting up monitoring and reporting collection of various device parameters such as CPU, memory usage, battery usage, rendering time, network data usage etc. requires use of various tools many of which are supplied by the mobile OS vendors along with the SDK.

To reduce the cost of load generation infrastructure and tool cloud supported tools may also be used.

## 5. Performance Testing solutions for different Mobile Applications 605 minutes (K3)

Terms: Monitor, Instruments, Jank, Memory Leaks, Heap, RRC, HTTP Pipelining, Caching

### 5.1 Common Performance Issues for Mobile Applications 20 minutes (K1)

*PTFL 5.1-1 Recall common performance issues for Mobile Applications (K1)*

#### 5.1.1 Common Performance Issues

Some of the common performance issues faced by a large number of mobile applications are

- Large amount of data transfers
- Large storage requirements
- App being battery unfriendly
- Memory leaks and resource hogging applications
- Slow response time for startup and various actions
- Graceless handling of unexpected situations like network outage and others
- Server side performance issues

### 5.2 Native App Performance Testing 210 (Android)+120 (iOS) minutes (K3)

*PTFL 5.2-1 Be able to understand the reasons of performance issues for native apps (K2)*

*PTFL 5.2-2 Be able to monitor performance for Android and iOS native apps using various tools (K3)*

#### 5.2.1 Common Performance Issues

The native apps face various performance challenges because of weaker mobile CPU, lesser memory and storage space availability. Network issues too need to be taken into consideration. Power consumption is almost never an issue for common desktop/web applications but it is a very important factor for mobile devices because of limited battery power capacities.

There are a variety of sensors available on the devices which need to be handled appropriately and their power consumption also needs to be taken into account when designing an application.

#### 5.2.2 Common Tools and Indicators to be monitored

Android IDE such as Android Studio comes bundled with a monitoring tool called Monitor and Android Device Monitor and Xcode for iOS comes with a tool called Instruments. These tools, in addition to other third party tools can be used to monitor various device side parameters such as the Battery, the Display, various network related statistics and various resources such as the CPU, Threads, Memory, Storage etc.

For Android devices understanding Wakelock behavior and monitoring apps for their appropriate use is

critical to well-performing apps. ADB (Android Debug Bridge) provides various commands such as `dumpsys` for getting access to various types of logs for e.g., battery stats, CPU info etc., created on the device during normal operations and these logs can be used to analyze the app performance behavior.

Some of the external tools used are those built-in such as battery status on the device or external tools such as the battery historian

An App in iOS moves through various states and at each transition various resource usage parameters need to be checked. Instruments tool can be used to monitor various parameters and find out amongst other things, memory leaks, heap allocations, power profiles etc.

### 5.3 Mobile Web App Performance Testing 130 (Android) + 30 (iOS) minutes (K3)

***PTFL 5.3-1 Be able to monitor performance for Android and iOS Web apps using various tools (K3)***

Web applications including Mobile Web application rely on the browser on the desktop/device. Some applications use a mobile specific version of a website whereas others use responsive design and may send different content based on the user agent identification string of the website. In all the cases the client side performance depends on some common things such as Radio Resource Control, HTTP Pipelining, Browser Cache and JavaScript execution. Concepts like HTTP pipelining and waterfall need to be understood to analyze the reasons for poor client-side performance. In addition there are tools like those that come with browser (Dev tools) such as chrome and safari. For Android ARO is an example of a tool that provides detailed analysis of performance.

### 5.4 Mobile Network Performance 20 minutes (K1)

***PTFL 5.4-1 Be able to recall use of various tools for network emulation (K1)***

Network emulation allows users to simulate/emulate various network conditions. There are many type of emulators available, some of them being every comprehensive and powerful but difficult to setup and others which are easy to setup and are part of various load testing applications such as LoadRunner and NeoLoad.

### 5.5 Server Side Performance Testing 30 minutes (K3)

***PTFL 5.5-1 Be able to understand the method of performance testing server side (K2)***

***PTFL 5.5-2 Be able to use at least one tool for load testing the server side (K3)***

#### **5.5.1 Method and Setup**

Server Side performance testing requires monitoring various parameters on the server side. The tools used for loading the server side require capturing network traffic either on the device or using a proxy on the desktop. The difference between Web app and Mobile app performance testing is that the traffic to be captured comes from mobile devices rather than web browsers.

### **5.5.2 Monitoring and Analysis**

Various parameters on the server side need to be monitored and later analyzed for bottleneck identification. Some of them are - Processor, Memory, Network I/O, Disk I/O and various OS/Application Specific parameters. Once the data is collected it needs to be analyzed to understand the bottlenecks.

Various tools like JMeter, LoadRunner, NeoLoad etc. can be used for server side performance testing of mobile applications.

**Note: Spend 60 minutes on any one of the tools.**