

**PROJECT REPORT  
ON**

**ANALYSIS OF PRACTO HEALTHCARE PLATFORM**

*A Case-Based Comparative Find out of Software  
Development Life Circle Models in Healthcare*

**Submitted to  
NMAM Institute of Technology, Nitte  
(Off-Campus Centre, Nitte Deemed to be University,  
Nitte - 574 110, Karnataka, India)**

**In partial fulfillment of the requirements for the  
award of the  
Degree of Bachelor of Technology  
in  
Information Science and Engineering  
By**

Anwesh Kumar NNM24IS037

**Under the guidance of  
Dr. JASON ELROY MARTIS  
Associate Professor**



## **Introduction**

## **Background**

Healthcare sector is one of the most sensitive and important industries in the world. Earlier, most hospital activities were done manually like appointment booking, maintaining patient records, billing, pharmacy management and so on. This manual process created delays, confusion and sometimes even mistakes.

With the growth of digital technology, many healthcare platforms started providing online services. Practo is one such leading healthcare platform in India which connects patients, doctors, clinics and hospitals through a single digital system.

### **Practo allows users to:**

1. Search doctors based on specialization
2. Book appointments
3. Do online video consultations
4. Sort medicines
5. Book lab tests
6. Rescue digital health records [3]

The system handles very sensitive data like patient medical history, prescriptions, payment information and personal details. Because of this, software development for Practo must follow proper Software Development Life Round (SDLC) practices.[1]

This project studies Practo's platform from SDLC perspective and analyzes which development model is most suitable.

## **Objectives**

**The main objectives of this report are:**

- To check Practo healthcare system using SDLC concepts
- To look at Waterfall, Incremental and Boehm' Spiral models
- To get how requirements engineering plays major role
- To find out risk management in healthcare systems
- To check Agile, DevOps and emerging technologies
- To get put simplified requirements document

## **Scope**

**This report includes:**

- Overview of Practo healthcare platform
- Comparison of three SDLC models
- Requirements Engineering process
- Requirements validation strategy
- Discussion of risks and challenges
- Combination of AI, IoT and Blockchain
- Consideration of Agile and DevOps

## **Methodology**

The methodology used in this find out includes:

- Literature check of SDLC models[1]
- Case find out analysis of Practo
- Comparison of suitability of models
- Preparation of requirements document
- Find out of real-world healthcare challenges

# Comparison and Analysis of Software Process Models

## Incremental Development Model

The Incremental Model develops software in small parts instead of building everything at once. Each increment adds new features to the system.

### Suitability for Practo Functional Requirements

In Practo's case, the development can be done in stages:

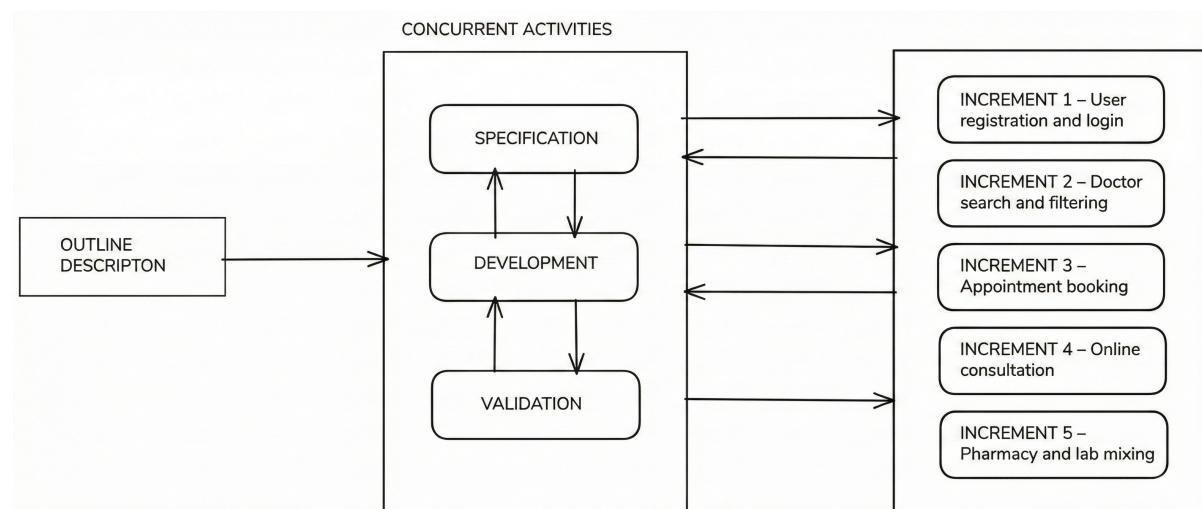
Increment 1 – User registration and login

Increment 2 – Doctor search and filtering

Increment 3 – Appointment booking

Increment 4 – Online consultation

Increment 5 – Pharmacy and lab mixing



This approach makes sense because Practo didn't start with all features in the beginning. Over time, new services were added.

## **Non-Functional Requirements**

Security, performance and scalability can also be improved step by step. For example:

- First version may support limited users
- Later updates make improve server optimization
- Additional security layers added gradually

## **Risk and Move Management**

Healthcare requirements go frequently. During COVID-19, teleconsultation demand increased suddenly. Incremental model supports such quick changes.

If any issue happens in one module, it doesn't break the whole system.

## **Time and Cost Constraints**

Incremental approach allows early let go of core features. Revenue can start early instead of waiting for full system completion. This reduces financial risk.

### **Advantages**

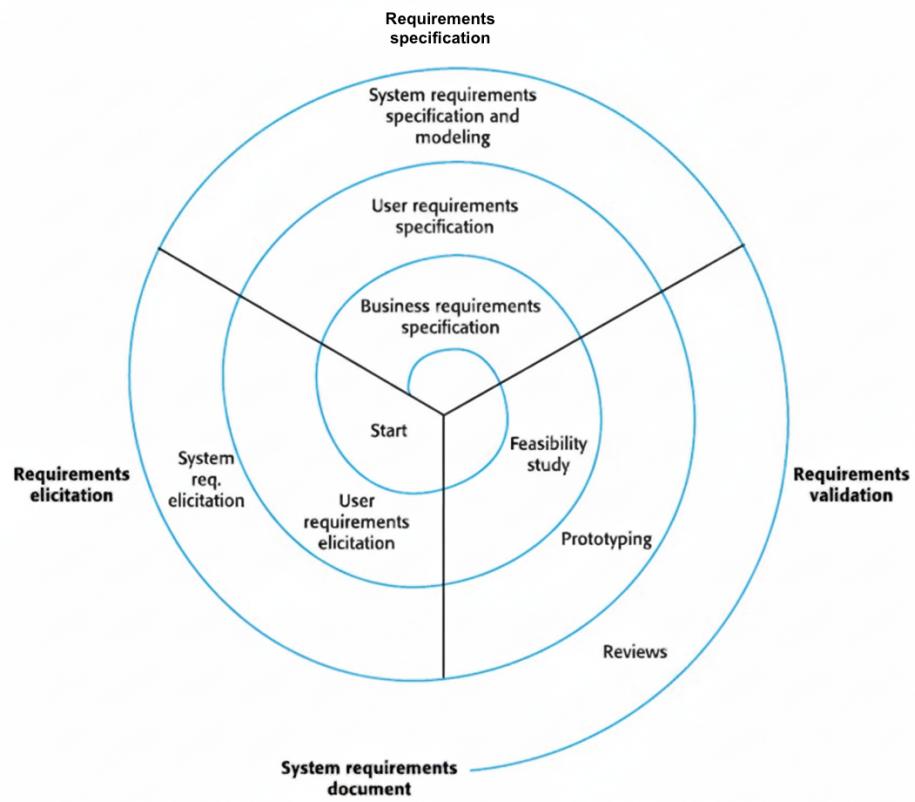
- Flexible to requirement changes
- Faster delivery of useful features
- Easy testing and debugging

### **Disadvantages**

- Combination of many modules may become complex
- Needs proper planning
- Some rework may be required later

Overall, Incremental model is highly suitable for Practo.

## Boehm's Spiral Model



The Spiral Model was proposed by Barry Boehm [2]. Boehm's Spiral Model focuses mainly on risk analysis. It combines iterative development with continuous risk assessment.[2][6]

## Suitability for Practo

### Functional Requirements

Since Practo deals with medical records and payment data, risk management becomes very important. Boehm's Spiral model ensures each feature is tested for legal and privacy risks.

## **Non-Functional Requirements**

Performance and security risks are evaluated at every circle. For example:

- Risk of data leakage
- Risk of payment fraud
- Risk of server crash during peak hours

Each of these can be analyzed before full deployment.

## **Risk and Go Management**

Boehm' Spiral model is best when risk is high.

Healthcare sector always involves risk because wrong handling of data may go patient safety.

## **Time and Cost Constraints**

**Boehm' Spiral model may require more cost because:**

- Risk analysis needs experts
- Prototyping increases time
- More documentation required

But in long run, it reduces cost of failure.

## **Advantages**

- Strong risk management
- High quality output
- Continuous improvement

## **Disadvantages**

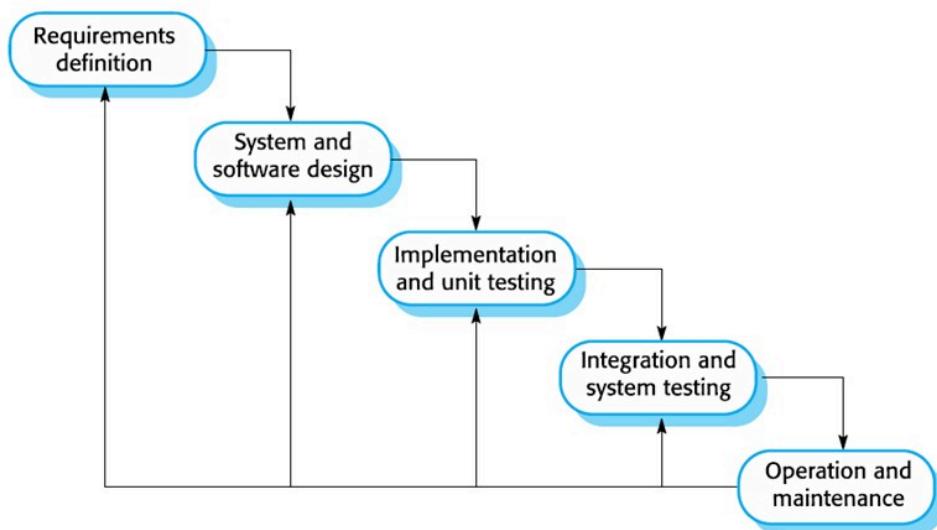
- Expensive
- Time consuming
- Needs experienced professionals

Turn is very suitable for core modules like digital health records and online payment system of Practo.[6][1]

## **Waterfall Model**

Waterfall model is a linear approach where each phase must complete before next phase begins [1].

**Waterfall Model follows a linear sequence:**



Each phase must complete before next phase starts.

## **Suitability for Practo**

### **Functional Requirements**

Waterfall works best when requirements are clearly defined and stable. For example, basic patient registration system might not go much.

### **Non-Functional Requirements**

Documentation is strong in Waterfall. For compliance-based industries, this is useful.

### **Risk and Go Management**

Once development begins, changes are difficult. Practo operates in active environment. So frequent go is common.

### **Time and Cost Constraints**

If requirements go after implementation phase, cost increases a lot.

## **Advantages**

- Clear structure
- Easy to get
- Proper documentation

## **Disadvantages**

- Not flexible
- Late detection of errors
- Slow response to new market demand

Waterfall may not be ideal for entire Practo system. But can be used for small stable components.

## **Summary Comparison of the Three Models**

	A	B	C	D	
1	Aspect	Waterfall	Incremental	Spiral	
1	Approah	Linear	In parts	Iterative with risk analysis	
2	Flexibility	Low	Medium to High	High	
3	Risk Management	Low	Medium to High	High	
4	User Involvement	Low	Moderate	High	
5	Cost	At first low	Moderate	High	
6					
7					

From comparison, Incremental + Spiral combination +Waterfall seems most suitable for Practo.[6][2][1]

## **Requirements Engineering Process**

Requirements Engineering is the backbone of any software project. If requirements aren't properly gathered, the system may do badly even if coding is perfect.

## **Simplified Requirements Document for Practo**

### **Functional Requirements**

#### **1. Patient Registration**

System should allow new patient registration and secure login.

#### **2. Doctor Registration**

Doctors must upload credentials for verification.

### **3. Appointment Scheduling**

Users must view available time slots and book appointments.

### **4. Online Consultation**

System should support video and chat consultation.

### **5. Digital Prescriptions**

Doctors should upload prescriptions digitally.

### **6. Online Payments**

Secure payment gateway must be integrated.

### **7. Lab Test Booking**

Patients should schedule home sample collection.

### **8. Medicine Ordering**

Users can sort prescribed medicines online.

### **9. Admin Dashboard**

Admin should manage users, reports and system monitoring.

# **Non-Functional Requirements**

## **1. Security**

Data must be encrypted and comply with healthcare standards.

## **2. Performance**

System must support at least 50,000 concurrent users.  
[6][1]

## **3. Scalability**

Should support future expansion to more cities and countries.

## **4. Reliability**

System uptime should be 99.9%.

## **5. Usability**

Interface should be simple even for elderly users.

## **6. Compatibility**

Should work on Android, iOS and Web platforms.

## **Requirements Elicitation Techniques**

- Interviews with doctors and patients
- Online surveys for user feedback
- Workshops with stakeholders
- Observation of hospital workflow
- Find out of legal healthcare documents.[1]

## **Validation Strategy**

**To ensure requirements are correct:**

- Conduct stakeholder check meetings
- Get bigger working prototypes
- Use test case mapping
- Make Requirements Traceability Matrix[1]

## **Requirements Challenges**

1. Different expectations of doctors and patients
2. Strict healthcare data privacy laws
3. Changing government regulations
4. Budget limitations
5. Technical mixing issues

## **Overcoming Challenges**

- Continuous stakeholder communication
- Regular security testing
- Clear documentation
- Agile collaboration practices
- Expert consultation for compliance

## **Expanding SDLC Considerations**

### **Agile Methodology in Practo**

#### **Agile helps in:**

- Faster feature let go
- Continuous feedback
- Improve adaptability
- Quick bug fixes

But it needs constant involvement from stakeholders.[4]

# **DevOps Integration**

**DevOps allows:**

- Continuous cobination
- Continuous deployment
- Faster updates
- Reduced downtime

In healthcare platform like Practo, downtime can go patient services. So DevOps is very helpful.[5][4]

## **Ethical Considerations**

**Healthcare platforms must ensure:**

- Patient consent
- Data privacy
- Equal access to service
- Ethical use of AI

Ignoring ethics may make legal and social problems.[6]

## **Mixing with Emerging Technologies**

### **Artificial Intelligence**

**AI in Practo can:**

- Predict disease patterns
- Assist doctors in diagnosis
- Automate appointment reminders
- Find abnormal reports[4]

### **Internet of Things (IoT)**

- Wearable health devices
- Remote patient monitoring
- Automatic alert system
- Secure medical record sharing
- Improved trust among stakeholders
- Tamper-proof data storage
- Alternative SDLC Models
- Prototyping Model

Useful for testing UI design and patient interaction screens.[5]

**Advantage:**

Early feedback.

**Disadvantage:**

Can increase cost if overused.

**V-Model:**

Strong focus on validation and verification.

Useful when compliance and quality are critical.

The V-Model is an extension of the Waterfall. model but with strong emphasis on Verification and Validation. So each development phase has a corresponding testing phase, and So it ensures high quality and compliance.

**Conclusion:**

After detailed analysis, it's clear that:

Waterfall is suitable only for small stable modules

Incremental is good for feature expansion

Spin is best for managing risk

So, for Practo healthcare platform, Hybrid approach combining Incremental and Spin model is most practical.[2]

Healthcare software must focus not only on functionality but also on security, reliability and compliance. Any small mistake can make serious consequences.

## **By applying strong requirements**

engineering process and combining Agile and DevOps practices, Practo can rescue high quality healthcare services digitally.

## **Appendix**

### **Tools used in Requirements Management:**

- JIRA
- Trello
- IBM DOORS

**Github Repository link:** <https://github.com/Anwesh004/Practo-SDLC-Analysis>

## **References**

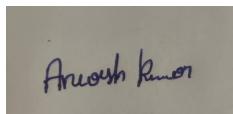
1. I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson, 2016.

2. B. W. Boehm, “A spiral model of software development and enhancement,” *Computer*, vol. 21, no. 5, pp. 61–72, May 1988, doi: 10.1109/2.59.
3. Practo, “Healthcare platform services,” Practo.com. Accessed: Feb. 12, 2026. [Online]. Available: <https://www.practo.com/about>
4. B. Meskó, G. Hetényi, and B. Győr, “Will artificial intelligence solve the human resource crisis in healthcare?,” *BMC Health Services Research*, vol. 18, no. 545, Jul. 2018, doi: 10.1186/s12913-018-3352-4.
5. S. Jalali and C. Wohlin, “Global software engineering and agile methods: A systematic review,” *J. Software: Evolution and Process*, vol. 24, no. 6, pp. 643–659, Aug. 2012, doi: 10.1002/smrv.567.
6. A. Kumar and R. Singh, “Digital health platforms in India: A case study of Practo,” *J. Healthcare Informatics Research*, vol. 7, no. 2, pp. 112–130, 2023, doi: 10.1007/s41666-023-00145-7.
7. I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.
8. B. W. Boehm, “A spiral model of software development and enhancement,” *Computer*, vol. 21, no. 5, pp. 61–72, May 1988, doi: 10.1109/2.59.
9. Practo, “Healthcare platform services,” Practo.com. Accessed: Feb. 12, 2026. [Online]. Available: <https://www.practo.com/about>
10. B. Meskó, G. Hetényi, and B. Győr, “Will artificial intelligence solve the human resource crisis in healthcare?,” *BMC Health Services Research*, vol. 18, no. 545, Jul. 2018, doi: 10.1186/s12913-018-3352-4.

- 11.S. Jalali and C. Wohlin, “Global software engineering and agile methods: A systematic review,” *J. Software: Evolution and Process*, vol. 24, no. 6, pp. 643–659, Aug. 2012, doi: 10.1002/smri.567.
- 12.A. Kumar and R. Singh, “Digital health platforms in India: A case study of Practo,” *J. Healthcare Informatics Research*, vol. 7, no. 2, pp. 112–130, 2023, doi: 10.1007/s41666-023-00145-7.

**DECLARATION OF ACADEMIC INTEGRITY:** I hereby declare that this assignment is my original work and hasn't been copied or plagiarized from any source. But all references and resources used have been properly cited, and Any ambiguities will lead to forfeiting my marks.

Student Signature



Submission date

12 February 2026