

MAKALAH
LEAN SOFTWARE DEVELOPMENT (LSD)



KELOMPOK 3

FANISA VIOLENA	(13020230120)
TIARA AFRINTAN RAMADHANI	(13020230127)
MUTIARA ABDI	(13020230131)
WA ODE RESKY AULIA A.	(13020230137)
NABILA MAHDIYA SAMAD	(13020230163)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR

2025

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan penyusunan makalah ini dengan judul Lean Software Development (LSD) tepat pada waktunya. Makalah ini disusun sebagai pemenuhan tugas mata kuliah/bidang studi Rekayasa Perangkat Lunak Objek prodi Teknik Informatika dan bertujuan untuk mengkaji secara mendalam mengenai Lean Software Development (LSD) Kami menyadari bahwa dalam penyusunan makalah ini masih terdapat banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun dari pembaca sangat kami harapkan demi perbaikan di masa mendatang. Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam proses penyusunan makalah ini. Semoga makalah ini dapat memberikan kontribusi yang berarti serta menambah wawasan dan pengetahuan bagi para pembaca.

Makassar, 27 September 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan Penulisan	2
BAB II	3
1.1. Definisi dan Sejarah.....	3
1.2. Penerapan Model	4
1.3. Karakteristik model	4
1.4. Kelebihan dan Kelemahan	5
1.5. Perbedaan.....	5
1.6. Alat Bantu Model	6
BAB III	9
1.1. Kesimpulan	9
DAFTAR PUSTAKA.....	10

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam lanskap pengembangan perangkat lunak modern yang terus berkembang, organisasi dihadapkan pada tekanan yang semakin besar untuk menghadirkan produk berkualitas tinggi dengan lebih cepat dan efisien. Pencarian metodologi yang dapat mengoptimalkan proses, mengurangi pemborosan, dan mempercepat waktu pengiriman tanpa mengorbankan kualitas menjadi prioritas utama. Secara historis, model pengembangan perangkat lunak tradisional seperti Waterfall, meskipun menawarkan struktur yang jelas dan mudah dipahami, seringkali menunjukkan keterbatasan signifikan. Keterbatasan ini terutama terlihat dalam kemampuannya menghadapi perubahan persyaratan yang dinamis, yang merupakan karakteristik umum dalam proyek perangkat lunak saat ini. Umpan balik pelanggan yang terlambat dalam model Waterfall dapat menyebabkan risiko kegagalan proyek yang tinggi dan kurangnya fleksibilitas untuk beradaptasi dengan kebutuhan pasar yang berubah.

Menanggapi tantangan ini, pendekatan Agile muncul sebagai alternatif yang lebih adaptif dan responsif. Agile menekankan pengembangan iteratif dan inkremental, kolaborasi tim, dan umpan balik berkelanjutan, yang terbukti lebih efektif dalam lingkungan yang tidak pasti. Di antara berbagai metodologi Agile, Lean Software Development (LSD) menonjol sebagai pendekatan yang berakar kuat dari prinsip-prinsip Lean Manufacturing, yang dipelopori oleh Toyota Production System. Tujuan utama LSD adalah memaksimalkan nilai pelanggan sambil secara sistematis meminimalkan segala bentuk pemborosan dalam proses pengembangan perangkat lunak.

LSD secara fundamental berfokus pada penghapusan fitur yang tidak perlu, pengurangan waktu tunggu yang tidak produktif, peningkatan alur kerja yang mulus, dan pemberdayaan tim pengembangan untuk mengambil keputusan yang relevan. Pendekatan ini telah terbukti secara empiris efektif dalam mengurangi waktu pengerjaan proyek, meningkatkan aliran nilai dari ide hingga produk jadi, dan secara signifikan memperbaiki tingkat perbaikan cacat. Dengan demikian, LSD menawarkan kerangka kerja yang kuat untuk mencapai efisiensi operasional dan keunggulan produk dalam industri perangkat lunak yang kompetitif.

Makalah ini akan menguraikan secara komprehensif mengenai Lean Software Development. Pembahasan akan dimulai dari definisi dan sejarahnya, dilanjutkan dengan analisis mendalam tentang penerapan model, karakteristik utama termasuk struktur tim dan aliran proses, serta kelebihan dan kelemahan yang melekat pada metodologi ini. Selain itu,

makalah ini akan membandingkan LSD dengan metodologi pengembangan perangkat lunak lainnya, seperti model Waterfall, Agile, dan Scrum, untuk menyoroti perbedaan dan sinergi mereka. Bagian penting lainnya adalah identifikasi alat bantu atau teknologi yang digunakan dalam penerapan LSD. Sebagai fokus utama, studi kasus aplikasi akan disajikan untuk memberikan pemahaman yang lebih mendalam dan ilustrasi praktis tentang bagaimana prinsip-prinsip LSD diterapkan dalam skenario pengembangan perangkat lunak di dunia nyata, memberikan wawasan tentang tantangan dan keberhasilan implementasinya.

1.2. Tujuan Penulisan

Tujuan dari penulisan makalah ini adalah:

- a. Menguraikan definisi dan sejarah Lean Software Development (LSD).
- b. Menjelaskan penerapan model LSD dalam siklus pengembangan perangkat lunak.
- c. Mengidentifikasi karakteristik utama model LSD, termasuk tim dan aliran proses.
- d. Menganalisis kelebihan dan kelemahan penerapan LSD.
- e. Membandingkan LSD dengan model pengembangan perangkat lunak lainnya, khususnya model Waterfall, serta Agile dan Scrum.
- f. Mengidentifikasi alat bantu atau teknologi yang digunakan dalam penerapan LSD.

BAB II

KAJIAN TEORITIS

1.1. Definisi dan Sejarah

Lean Software Development (LSD) adalah sebuah metodologi pengembangan perangkat lunak yang berakar kuat dari prinsip-prinsip Lean Manufacturing, sebuah filosofi produksi yang pertama kali dikembangkan oleh Toyota Production System di Jepang. Konsep "Lean" dalam konteks pengembangan perangkat lunak pertama kali diperkenalkan secara formal oleh Mary dan Tom Poppendieck dalam buku seminal mereka, "Lean Software Development: An Agile Toolkit," yang diterbitkan pada tahun 2003. Buku ini secara eksplisit menghubungkan prinsip-prinsip Lean dengan metodologi pengembangan perangkat lunak Agile yang sedang berkembang pesat.

Definisi LSD berpusat pada dua pilar utama: penciptaan nilai maksimal bagi pelanggan dan penghapusan segala bentuk pemborosan (waste) dalam setiap aspek proses pengembangan perangkat lunak. Pemborosan, dalam konteks LSD, adalah segala aktivitas atau hasil yang tidak menambah nilai bagi produk akhir atau pelanggan. Ini bisa berupa fitur yang tidak perlu (over-production), penundaan dalam alur kerja (waiting), pengerjaan ulang karena cacat (defects), tugas yang sebagian selesai (partially done work), transfer informasi yang tidak efisien (motion), pemrosesan yang berlebihan (over-processing), dan bahkan aktivitas manajemen yang tidak perlu. Tujuan utama LSD adalah mengoptimalkan aliran nilai (value stream) secara menyeluruh, mulai dari ide awal hingga pengiriman produk akhir yang berfungsi kepada pelanggan.

Sejarah LSD tidak dapat dilepaskan dari kebutuhan industri perangkat lunak untuk menjadi lebih efisien dan efektif dalam menghadapi tantangan pasar yang semakin kompleks dan kompetitif. Seiring dengan krisis ekonomi global pada tahun 2009, banyak perusahaan mulai mencari cara untuk membuat upaya pengembangan perangkat lunak mereka lebih efisien, dan prinsip-prinsip lean dari manufaktur menjadi inspirasi. LSD muncul sebagai respons terhadap keterbatasan metodologi tradisional, seperti Waterfall, yang seringkali kaku dan tidak adaptif terhadap perubahan persyaratan. Bahkan, LSD juga menjadi respons terhadap beberapa praktik Agile yang, meskipun lebih fleksibel, terkadang terlalu fokus pada jangka pendek tanpa mempertimbangkan optimasi aliran nilai secara holistik. Dengan demikian, LSD menawarkan kerangka kerja filosofis yang lebih dalam untuk mencapai kelincahan dan efisiensi yang berkelanjutan dalam pengembangan perangkat lunak.

1.2. Penerapan Model

Penerapan model Lean Software Development didasarkan pada tujuh prinsip utama:

- a. **Menghilangkan Pemborosan (Eliminate Waste):** Mengidentifikasi dan menghilangkan aktivitas yang tidak menambah nilai bagi pelanggan, seperti fitur yang tidak perlu, dokumentasi berlebihan, dan penundaan.
- b. **Memperkuat Pembelajaran (Amplify Learning):** Mendorong siklus umpan balik yang cepat dan pembelajaran berkelanjutan melalui eksperimen dan iterasi.
- c. **Mengambil Keputusan Selambat Mungkin (Decide as Late as Possible):** Menunda keputusan penting hingga informasi yang memadai tersedia, memungkinkan fleksibilitas dan adaptasi terhadap perubahan.
- d. **Mengirim Secepat Mungkin (Deliver as Fast as Possible):** Mempercepat waktu pengiriman perangkat lunak yang berfungsi untuk mendapatkan umpan balik lebih awal dan memberikan nilai kepada pelanggan dengan cepat.
- e. **Memberdayakan Tim (Empower the Team):** Memberikan otonomi dan tanggung jawab kepada tim pengembangan, mendorong pengambilan keputusan di tingkat tim.
- f. **Membangun Integritas (Build Integrity In):** Memastikan kualitas dan konsistensi produk melalui pengujian berkelanjutan dan praktik pengembangan yang baik.
- g. **Melihat Keseluruhan (See the Whole):** Memahami sistem secara keseluruhan dan mengoptimalkan aliran nilai dari ujung ke ujung, bukan hanya bagian-bagian individual. Siklus pengembangan produk lean melibatkan orientasi nilai, pengurangan biaya overhead, peningkatan antarmuka, dan pemberdayaan pengembang untuk mencapai peningkatan berkelanjutan.

1.3. Karakteristik model

- a. **Tim:** Tim dalam LSD cenderung otonom dan diberdayakan untuk membuat keputusan. Anggota tim Lean berusaha meminimalkan semua dokumentasi untuk menghemat waktu dan mengurangi pemborosan. Kolaborasi dan komunikasi yang efektif sangat ditekankan.
- b. **Aliran Proses:** Aliran proses dalam LSD berfokus pada kelancaran dan efisiensi. Ini melibatkan sistem "Pull" di mana pekerjaan ditarik ke depan hanya ketika ada permintaan, bukan "Push" di mana pekerjaan didorong ke tahap berikutnya terlepas dari kesiapan. Pemetaan aliran nilai (Value Stream Mapping) dan sistem Kanban sering digunakan untuk memvisualisasikan dan mengoptimalkan aliran kerja. Tujuannya adalah untuk mengurangi waktu tunggu dan memastikan pengiriman yang berkelanjutan dan mulus kepada pelanggan.

1.4. Kelebihan dan Kelemahan

Kelebihan:

- a. **Pengurangan Waktu Pengerjaan (Reduced Lead Time):** Salah satu manfaat paling umum yang dilaporkan adalah pengurangan waktu pengerjaan, yang berkontribusi pada peningkatan alur dan pengiriman yang lebih cepat.
- b. **Peningkatan Kualitas dan Produktivitas:** Dengan fokus pada penghapusan pemborosan dan pembelajaran berkelanjutan, LSD dapat meningkatkan kualitas produk dan produktivitas tim.
- c. **Fleksibilitas dan Adaptasi terhadap Perubahan:** LSD mendorong toleransi terhadap perubahan dan kemampuan untuk beradaptasi dengan persyaratan pelanggan yang berubah.
- d. **Peningkatan Nilai Pelanggan:** Dengan berfokus pada fitur yang benar-benar bernilai bagi pelanggan, LSD memastikan bahwa upaya pengembangan diarahkan pada hal yang paling penting.
- e. **Pengurangan Biaya:** Penghapusan pemborosan secara langsung berkontribusi pada pengurangan biaya pengembangan.

Kelemahan:

- a. **Resistensi terhadap Perubahan:** Adopsi pola pikir Lean bisa menjadi tantangan karena resistensi terhadap cara kerja baru.
- b. **Identifikasi "Pemborosan Sejati":** Sulit untuk mengidentifikasi apa yang sebenarnya merupakan pemborosan dalam organisasi dan bahkan lebih menantang untuk menghilangkannya.
- c. **Tantangan Skala Fleksibilitas:** Meskipun pembentukan tim dan swa-organisasi tidak sulit dicapai, skala fleksibilitas dan keterlibatan tugas manajemen bisnis jauh lebih menantang dalam cara kerja lean.
- d. **Ekspektasi yang Berlebihan:** Terkadang, ekspektasi terhadap hasil LSD terlalu tinggi, dan prinsip-prinsip lean hanya diperkenalkan secara dangkal, yang mengarah pada hasil yang kurang memuaskan.

1.5. Perbedaan

Lean Software Development (LSD) vs. Waterfall Model:

- a. **Waterfall:** Model sekuensial linear di mana setiap fase harus diselesaikan sebelum fase berikutnya dimulai, tanpa tumpang tindih. Persyaratan didefinisikan dengan jelas di awal, dan perubahan sulit diakomodasi. Risiko kegagalan tinggi jika persyaratan berubah di tengah jalan. Cocok untuk proyek dengan persyaratan yang sangat stabil dan jelas.

- b. **LSD:** Metodologi adaptif dan iteratif yang berfokus pada pengiriman nilai berkelanjutan, penghapusan pemborosan, dan fleksibilitas terhadap perubahan. Umpan balik pelanggan sering diterima, dan keputusan penting ditunda. Lebih cocok untuk proyek dengan persyaratan yang berkembang dan lingkungan yang dinamis.

LSD vs. Agile (Scrum):

- a. **Agile:** Metode pengembangan perangkat lunak ringan yang berfokus pada pengiriman fitur perangkat lunak kecil kepada pelanggan secepat mungkin dalam iterasi singkat. Agile menekankan kolaborasi, interaksi individu, perangkat lunak yang berfungsi, dan respons terhadap perubahan.
- b. **Scrum:** Kerangka kerja Agile paling populer yang menggunakan model pengembangan iteratif dan inkremental, berfokus pada pengelolaan tugas dalam lingkungan pengembangan berbasis tim.
- c. **LSD:** Meskipun Lean dan Agile memiliki banyak kesamaan dan sering dikombinasikan, LSD dapat dianggap sebagai filosofi yang lebih luas yang mendasari Agile. LSD memberikan fokus yang lebih kuat pada penghapusan pemborosan dan optimasi aliran nilai, sementara Agile (dan Scrum) menyediakan kerangka kerja dan praktik spesifik untuk mencapai tujuan tersebut. Lean telah berperan penting dalam membentuk metodologi Agile, dengan prinsip-prinsip intinya seperti pengurangan pemborosan, optimasi proses, dan nilai pelanggan yang terjalin dalam berbagai metode Agile.

1.6. Alat Bantu Model

Beberapa alat bantu dan praktik yang digunakan dalam Lean Software Development (LSD) sangat penting untuk mendukung implementasi prinsip-prinsip Lean dan mencapai efisiensi serta nilai maksimal. Berikut adalah pengembangan dari alat-alat tersebut:

- a. **Value Stream Mapping (VSM):** Ini adalah teknik visual yang digunakan untuk memetakan seluruh aliran nilai, mulai dari permintaan pelanggan hingga pengiriman produk akhir. VSM membantu tim mengidentifikasi semua langkah dalam proses, baik yang menambah nilai maupun yang merupakan pemborosan. Dengan memvisualisasikan aliran ini, tim dapat melihat di mana penundaan terjadi, di mana ada pekerjaan yang tidak perlu, dan di mana peluang untuk perbaikan proses dapat ditemukan. Tujuannya adalah untuk mengoptimalkan aliran kerja dan mengurangi waktu pengerjaan.
- b. **Kanban:** Kanban adalah sistem visual yang digunakan untuk mengelola aliran

kerja dan membatasi jumlah pekerjaan yang sedang berjalan (Work In Progress/WIP). Papan Kanban menampilkan tugas-tugas dalam berbagai tahap pengembangan, memungkinkan tim untuk melihat status proyek secara real-time. Dengan membatasi WIP, Kanban membantu mencegah tim menjadi terlalu banyak pekerjaan, mengurangi waktu tunggu, dan meningkatkan fokus pada penyelesaian tugas. Ini mendorong aliran kerja yang lebih lancar dan efisien.

- c. **Gemba Walk:** Praktik "Gemba Walk" berasal dari filosofi Lean Manufacturing, di mana "Gemba" berarti "tempat yang sebenarnya". Dalam konteks LSD, ini berarti manajer atau anggota tim pergi ke tempat pekerjaan sebenarnya dilakukan (misalnya, ke tim pengembangan) untuk mengamati proses, berinteraksi dengan tim, dan secara langsung memahami masalah serta menemukan peluang perbaikan. Ini membantu dalam mendapatkan pemahaman yang lebih dalam tentang tantangan operasional dan mempromosikan pengambilan keputusan berbasis fakta.
- d. **5W2H (Who, What, When, Where, Why, How, How Much):** Ini adalah alat analisis masalah yang sistematis. Dengan mengajukan pertanyaan-pertanyaan ini, tim dapat menggali akar penyebab masalah dan memahami konteksnya secara menyeluruh.
 - **Who (Siapa):** Siapa yang terlibat atau terpengaruh?
 - **What (Apa):** Apa masalahnya atau apa yang terjadi?
 - **When (Kapan):** Kapan masalah itu terjadi atau kapan tindakan harus diambil?
 - **Where (Di mana):** Di mana masalah itu terjadi?
 - **Why (Mengapa):** Mengapa masalah itu terjadi? (Ini seringkali memerlukan analisis lebih dalam untuk menemukan akar penyebab).
 - **How (Bagaimana):** Bagaimana masalah itu terjadi atau bagaimana cara menyelesaikannya?
 - **How Much (Berapa Banyak):** Berapa biaya atau dampak dari masalah tersebut?
- e. **Pareto Analysis:** Analisis Pareto, yang didasarkan pada prinsip Pareto (aturan 80/20), digunakan untuk mengidentifikasi dan memprioritaskan masalah. Ini membantu tim fokus pada beberapa penyebab utama yang bertanggung jawab atas sebagian besar masalah atau cacat. Dengan membuat daftar masalah secara hierarkis berdasarkan frekuensi atau dampaknya, tim dapat mengalokasikan sumber daya secara efektif untuk mengatasi masalah yang paling signifikan terlebih dahulu.
- f. **8D (Eight Disciplines Problem Solving):** Ini adalah metode terstruktur untuk

mengoreksi efek masalah dan mengobati penyebabnya untuk memastikan tidak muncul kembali. Metode ini melibatkan delapan langkah disipliner yang membantu tim dalam mengidentifikasi, menganalisis, dan menyelesaikan masalah secara permanen. Langkah-langkahnya meliputi pembentukan tim, deskripsi masalah, implementasi tindakan penahanan, identifikasi akar penyebab, pengembangan tindakan korektif, implementasi tindakan korektif, pencegahan pengulangan, dan pengakuan tim.

- g. **Test-Driven Development (TDD):** TDD adalah praktik pengembangan perangkat lunak di mana pengembang menulis tes otomatis sebelum menulis kode produksi. Ini membantu mempertahankan desain yang baik, memastikan fungsionalitas yang benar, dan mengurangi cacat sejak dini dalam siklus pengembangan. Dengan TDD, kode yang dihasilkan cenderung lebih modular, mudah dipelihara, dan memiliki cakupan pengujian yang tinggi.
- h. **Continuous Integration/Continuous Delivery (CI/CD):** CI/CD adalah praktik yang mengotomatiskan proses integrasi dan pengiriman perangkat lunak.
 - **Continuous Integration (CI):** Pengembang secara teratur mengintegrasikan kode mereka ke repositori bersama, dan setiap integrasi diverifikasi oleh build dan tes otomatis. Ini membantu mendeteksi masalah integrasi lebih awal.
 - **Continuous Delivery (CD):** Perubahan kode secara otomatis dibangun, diuji, dan disiapkan untuk rilis ke lingkungan produksi. Ini mempercepat umpan balik dan pengiriman nilai kepada pelanggan, memungkinkan rilis yang lebih sering dan lebih kecil.

BAB III

PENUTUP

1.1. Kesimpulan

Lean Software Development (LSD) adalah metodologi pengembangan perangkat lunak yang berakar kuat dari prinsip-prinsip Lean Manufacturing, berfokus pada maksimalisasi nilai pelanggan dan eliminasi pemborosan di setiap tahapan siklus pengembangan. Dengan mengadopsi tujuh prinsip intinya—menghilangkan pemborosan, memperkuat pembelajaran, mengambil keputusan selambat mungkin, mengirim secepat mungkin, memberdayakan tim, membangun integritas, dan melihat keseluruhan LSD menawarkan kerangka kerja yang adaptif, efisien, dan berorientasi pada nilai. Implementasi prinsip-prinsip ini secara konsisten telah terbukti memberikan berbagai keunggulan signifikan, termasuk pengurangan waktu pengerjaan proyek, peningkatan kualitas produk, fleksibilitas yang lebih besar dalam menghadapi perubahan persyaratan, peningkatan nilai yang dirasakan oleh pelanggan, dan pengurangan biaya pengembangan secara keseluruhan.

Meskipun demikian, penerapan LSD tidak tanpa tantangan. Organisasi seringkali menghadapi resistensi terhadap perubahan budaya dan proses kerja yang baru. Selain itu, mengidentifikasi "pemborosan sejati" dalam konteks pengembangan perangkat lunak bisa menjadi kompleks, dan skalabilitas fleksibilitas dalam proyek yang lebih besar dapat menjadi hambatan. Ekspektasi yang tidak realistis atau implementasi prinsip-prinsip Lean yang dangkal juga dapat menghambat tercapainya hasil yang optimal.

Dalam perbandingan dengan metodologi lain, LSD menonjol karena sifatnya yang fleksibel dan responsif terhadap perubahan, sangat berbeda dari model Waterfall yang sekuensial dan kaku. Meskipun memiliki banyak kesamaan dan sering kali dikombinasikan dengan pendekatan Agile dan Scrum, LSD memberikan fondasi filosofis yang lebih luas, dengan fokus yang lebih mendalam pada optimasi aliran nilai dan penghapusan pemborosan. Berbagai alat bantu dan praktik seperti Value Stream Mapping (VSM), Kanban, Gemba Walk, 5W2H, Pareto Analysis, 8D, Test-Driven Development (TDD), dan Continuous Integration/Continuous Delivery (CI/CD) mendukung implementasi prinsip-prinsip LSD dalam praktik, memungkinkan tim untuk memvisualisasikan, mengelola, dan terus meningkatkan proses pengembangan mereka.

Secara keseluruhan, LSD merupakan pendekatan yang kuat dan relevan dalam industri perangkat lunak yang dinamis, memungkinkan organisasi untuk tidak hanya menghasilkan produk berkualitas tinggi tetapi juga melakukannya dengan cara yang lebih efisien dan adaptif terhadap kebutuhan pasar yang terus berubah.

DAFTAR PUSTAKA

- [1] A. Pinna, G. Baralla, M. Marchesi, "Raising Sustainability Awareness in Agile Blockchain-Oriented Software Engineering", *IEEE Xplore*, 2020.
- [2] F. Kiš's, B. Rossi, "Agile to Lean Software Development Transformation: a Systematic Literature Review", *ResearchGate*, 2018.
- [3] S. A. Al-Hassan, "Lean Software Development (LSD) Based On Lean Principles (LPS) and (LSD) Practices (Devops)", *Scribd*, 2021.
- [4] A. El Hachimi, A. E. Hachimi, "A case study of Lean IT: Leading a Lean project in software development projects", *Journal Template - IEEE-SEM*, 2021.
- [5] C. Ebert, P. Abrahamsson, N. V. Oza, "Lean Software Development", *IEEE Software*, 2012.
- [6] C. Ebert, "Lean Software Development", *ResearchGate*, 2013.
- [7] M. O' Connor, K. Conboy, D. Dennehy, N. Carroll, "A Comparative Case Study of Waterfall and Agile Management", *ResearchGate*, 2022.
- [8] R. Yadav, V. Yadav, M. L. Mittal, J. Yadav, "A Comparative study of Agile Software Development Methodology and traditional waterfall model", *IOSR Journal*, 2015.
- [9] T. Beck, "Lean Tools in Software Development: A Series Introduction", *Medium*, 2023.