



Lead Software Engineer

(known internally as Lead Engineer)

Role profile – October 2026

About The Open University

At The Open University, we believe that where you start in life should not limit where you can go. Established over 50 years ago, we are committed to making high-quality education accessible to anyone, anywhere. Through our unique approach to distance and online learning, we have transformed millions of lives by opening opportunities that might otherwise be out of reach.

Our mission is to be open to people, places, methods and ideas. As one of the world's leading providers of flexible learning, we continue to innovate and evolve, helping students develop the skills, knowledge and confidence to succeed throughout their lives and careers.

About Digital Services

Digital Services is at the heart of delivering life-changing learning experiences that enrich society. We combine trusted design, data expertise and modern technology to create, run and continuously improve the digital services that support students, staff and the wider University community.

We work in multidisciplinary teams to solve meaningful problems, deliver value iteratively and ensure that our services are secure, reliable and accessible. Whether supporting student journeys, enabling teaching and learning, or providing the technology foundations that power the University, our work has a direct impact on thousands of people every day.

Our objectives are to:

- Enhance the student and staff experience through simple, evidence-based solutions.
- Deliver value iteratively, securely and at pace, balancing innovation, quality and operational excellence.
- Foster a vibrant and supportive workplace that embraces continuous learning, growth and collaboration, empowering everyone to succeed.

As a Lead Software Engineer, you will play a key role in shaping and operating the digital products and platforms that enable the University's mission, helping teams deliver sustainable, resilient and valuable services for students and staff.

THE OPPORTUNITY

Be the technical co-owner of a digital product, partnering with the Product Manager to set direction, make informed trade-offs and deliver value for the University, students and staff.

What you will lead

- Own the product's end-to-end technical lifecycle, from discovery and architecture through build, deployment, live operation, evolution and retirement.
- Set technical vision and direction across architecture, software, infrastructure, environments, CI/CD, automated testing, security, observability, resilience and support.
- Remain hands-on: design, build, test, review, deploy and operate high-quality software, while tackling complex technical and production problems.
- Work with the Product Manager to explain technical feasibility, estimate work, expose dependencies and ensure non-functional requirements are addressed.
- Balance user and organisational outcomes with technical risk, cost, delivery speed, technical debt and long-term sustainability.
- Build operability, maintainability, accessibility, security, resilience and supportability into solutions from the outset.
- Lead the technical response to incidents, use evidence and operational insights to address root causes, and improve reliability.
- Create shared ownership by coaching engineers, enabling sound decisions and avoiding technical bottlenecks or single points of failure.
- Contribute to engineering standards, patterns, technical strategy and the wider engineering community of practice.
- Identify, evaluate and introduce emerging practices and tools, including the responsible use of AI to improve engineering outcomes.

How the role works:

You will work in close partnership with the Product Manager and Delivery Manager within a multidisciplinary product team. You will also report to and align with the Group Engineering Lead on engineering standards, capability and technical direction. This role provides technical coaching and mentoring, alongside formal line-management responsibility.

What you will bring

Essential experience:

These are the core requirements. We are looking for evidence that you have already operated with comparable technical scope and accountability.

- End-to-end technical ownership of a production digital product or service through discovery, delivery and live operation.
- Accountability for technical outcomes across architecture, software engineering, cloud infrastructure, CI/CD, automated testing, security, observability and resilience.
- Hands-on engineering experience sufficient to design solutions, write and review maintainable code, and influence implementation quality.
- Operating and supporting production services, including incident diagnosis, root-cause improvement and reliability work.
- Creating automated testing and safe, repeatable deployment pipelines as integral parts of software delivery.
- Making pragmatic architectural decisions and managing technical debt, vulnerabilities, obsolescence and lifecycle risk.
- Leading technical delivery in a multidisciplinary product team and working with Product to make evidence-based trade-offs.
- Coaching engineers and building collective ownership without becoming the team's decision-making bottleneck.

Desirable experience:

- Leading modernisation or migration of established production services.
- Contributing to engineering communities of practice, standards or reusable patterns beyond a single team.
- Using operational data, user feedback and delivery measures to drive continuous improvement.
- Evaluating and introducing new engineering tools or practices, including responsible AI-assisted development.

Technology landscape:

Depending on the team, you may work with: C# and .NET; Microsoft Azure; RESTful APIs and distributed systems; React and Next.js; headless CMS platforms; Terraform and Infrastructure as Code; GitHub and Azure DevOps; CI/CD; Playwright and other automated testing frameworks; containers and cloud-native technologies; monitoring, observability, automated security and quality tooling; and AI coding assistants such as GitHub Copilot.

You do not need experience of every technology listed. Strong engineering fundamentals, learning agility and evidence of applying modern engineering practices matter more than a precise technology match.

How you will work:

Lead through influence Create clarity and enable good decisions.	Be pragmatic Balance quality, value, risk, cost and time.	Stay outcome-oriented Focus on the problem and value created.
Build quality in Use automation and fast feedback.	Enable autonomy Grow shared ownership and capability.	Keep learning Challenge established practice constructively.

Candidate note:

Please use examples that show the scope of your accountability, the decisions you made, the outcomes achieved and how you strengthened the team's capability.

Role purpose and accountability

The Lead Software Engineer is the technical co-owner of one or more digital products or services. Working with the Product Manager, the role helps establish direction, understand constraints, make informed investment choices and ensure that the product delivers sustainable value. The Lead Software Engineer is accountable for technical direction, quality, health and sustainability throughout the full lifecycle.

The role is deliberately broader than application development alone. Accountability spans the technical system needed to deliver and operate the product effectively, including architecture, software, infrastructure, environments, configuration, delivery pipelines, automated testing, security, observability, resilience and operational support.

Working relationships

- **Product Manager:** Partner on product direction and trade-offs. Explain feasibility, estimates, dependencies, non-functional needs, risks and the implications of technical options.
- **Delivery Manager:** Work within the team's delivery approach. The Delivery Manager supports planning, flow and continuous improvement, enabling product and technical leadership to remain clear.
- **Group Engineering Lead:** You will report to the Group Engineering Lead and align on engineering standards, capability, technical direction and cross-team concerns. Escalate material technical risks and contribute evidence from the product context.
- **Engineers and specialists:** Provide technical direction, coaching, challenge and line management. Draw on specialist and shared services where useful while retaining end-to-end accountability for how the product works.
- **Wider engineering community:** Contribute patterns, lessons, standards and practical improvements that are useful beyond the immediate team.

Detailed responsibilities

Product & technical co-ownership:

- Work with Product to understand the user and organisational problem before selecting a technical approach.
- Make the consequences of technical options visible, including feasibility, cost, risk, dependencies, delivery implications and future constraints.
- Help prioritise technical investment alongside changes that directly deliver user or business value.
- Contribute technical insight during discovery, delivery, operation and product evolution.

Architecture & technical direction:

- Establish a coherent technical vision and guide architectural decisions without introducing unnecessary complexity.
- Ensure decisions are evidence-based, proportionate, documented and understood by the team.
- Consider security, privacy, accessibility, resilience, performance, scalability, operability and maintainability as part of design.
- Identify cross-product dependencies and work with relevant colleagues to manage them.

Build, run & own:

- Own the end-to-end technical lifecycle from discovery and architectural inception to retirement.
- Ensure the team understands and can operate its product in production, with appropriate tooling, access, information and support arrangements.
- Retain accountability for testing, infrastructure, deployment and live operation while collaborating with specialists and shared services where appropriate.
- Reduce unnecessary hand-offs and strengthen feedback between development and live service operation.

Engineering quality & delivery:

- Remain actively involved in design, coding, test automation, review, deployment and technical problem-solving.
- Promote clean, maintainable and well-tested code, with proportionate use of TDD, BDD and other quality practices.

- Create automated, repeatable and safe CI/CD processes with fast feedback.
- Help break complex work into small, valuable increments while protecting essential non-functional quality.

Operational health & resilience:

- Use monitoring, observability, operational data and user feedback to understand product behaviour and health.
- Lead or coordinate the technical response to incidents affecting the product.
- Turn incident learning into changes that reduce recurrence or improve detection, diagnosis and recovery.
- Build resilience, supportability and recovery considerations into product design and change.

Technical health & lifecycle management:

- Make technical debt visible and manage it as part of product investment decisions.
- Identify and address vulnerabilities, obsolescence, unsupported components and other lifecycle risks.
- Ensure decisions made during delivery remain appropriate when the product is operating at scale and over time.
- Plan for evolution, modernisation and eventual retirement rather than treating go-live as the end state.

Team & community enablement:

- Coach and mentor engineers in technical practice, decision-making and ownership.
- Use pairing, code review, technical discussion and communities of practice to share knowledge.
- Develop team capability so that knowledge and responsibility are distributed rather than concentrated.
- Contribute reusable standards, patterns and learning across the wider engineering community.

Innovation & emerging practice:

- Scan the horizon for relevant engineering methods, tools and platform capabilities.

- Evaluate new practices against real product needs, risks and benefits rather than adopting technology for its own sake.
- Introduce improvements responsibly, including AI-assisted engineering where it can improve quality, flow or learning.
- Create safe opportunities for experimentation, feedback and evidence-based adoption.

Experience and evidence guide

The selection process will look for specific examples of accountability and outcomes.

Evidence area	What strong evidence may include
Whole-product ownership	A product or service taken from problem framing or discovery through build and into sustainable live operation.
Architecture & trade-offs	A consequential decision where options, constraints, risks and longer-term implications were made clear.
Build, run & own	Evidence that the engineering team retained responsibility after go-live and improved the service using operational learning.
Quality & automation	Automated testing and delivery practices established or materially improved, with an explanation of the outcome.
Operational leadership	A production incident or reliability challenge led through diagnosis, recovery and root-cause improvement.

Technical health	Technical debt, lifecycle, security or obsolescence risks prioritised and managed with Product.
Team enablement	Other engineers became more capable and autonomous, with knowledge and decisions spread across the team.
Wider influence	Standards, patterns, communities or practices improved beyond the immediate product team.
Innovation	A new tool or practice evaluated and introduced responsibly, with benefits and limitations understood.

Ways of working

- **Lead through influence:** Create clarity, facilitate productive technical discussion and help others succeed.
- **Be pragmatic:** Recognise that there is rarely a perfect solution and make trade-offs visible.
- **Stay outcome-oriented:** Connect technical choices to the user, organisational or service outcome.
- **Value continuous delivery:** Prefer small increments, automation and fast feedback over large releases and hand-offs.
- **Build quality in:** Treat testing, security, accessibility, performance and reliability as continuous engineering concerns.
- **Enable autonomy:** Create conditions in which engineers can make sound decisions without centralising ownership.
- **Work with ambiguity:** Help discover what should be built rather than waiting for a complete specification.
- **Keep learning:** Explore new approaches and challenge established practice when evidence suggests a better outcome.

What you will bring

The technology mix varies by team. Current examples named in the role profile include:

- C# and .NET
- Microsoft Azure
- RESTful APIs and distributed systems
- React and Next.js
- Headless CMS platforms
- Terraform and Infrastructure as Code
- GitHub and Azure DevOps
- CI/CD pipelines
- Playwright and other automated testing frameworks
- Containerisation and cloud-native technologies
- Application monitoring and observability
- Automated security and quality tooling
- AI coding assistants such as GitHub Copilot

Candidates are not expected to match every technology. Assessment will focus on engineering fundamentals, learning agility and the ability to apply sound practices in context.

Role boundaries and clarifications

- This is a hands-on technical leadership role with whole-product accountability. It is broader than a role focused primarily on coding or delivery coordination.
- The role provides technical coaching and mentoring alongside formal line-management responsibilities.
- Specialists and shared services may contribute, but their involvement does not remove the Lead Software Engineer's accountability for the effective end-to-end technical outcome.
- The Product Manager retains primary accountability for product vision, strategy, user needs, prioritisation and value; technical and product decisions are made in close partnership.
- The Delivery Manager supports effective planning, flow and improvement, rather than replacing product or technical ownership.

