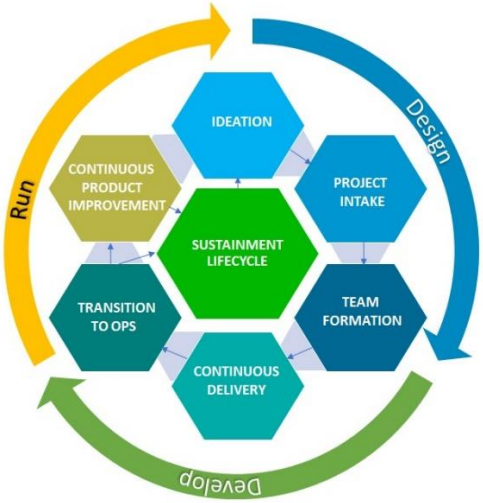
What does it do?

What are the benefits?

The Plays...

- 1. Ideate solutions that meet the needs of business
- 2. Successfully navigate the project intake process
- 3. Getting started to develop your solution
- 4. Developing a product
- 5. Transition to Ops
- 6. Continuous Product Improvement
- 7. Sustainment Lifecycle
- Definition of a Modern Application `



CITZ Modern Application Checklist

Principles

Patterns

Glossary

Released: June 12, 2020 *Feedback is appreciated*

Modern Application Playbook

This playbook is designed to help you on your IT project journey. It provides the reader with a collection of references to tested methodologies, tools and best practices. Developing and supporting government applications continues to shift from a requirements heavy traditional waterfall methodology towards a user centric agile approach with shorter project development cycles and the adoption of a DevOps culture that has been successfully pioneered within the BC Government by the BC Dev Exchange.

What does it do?

Contributes to a shared understanding of the solution ecosystem from ideation to sustained operating IM/IT systems. “Plays” describe valuable patterns that project teams should consider as part of their modern digital product management lifecycle.

What are the Benefits?

Provides the IMB and our partners, vendors and clients with key resources that will assist them to identify, use and leverage trustworthy sources of knowledge that is regularly improved with the aim of enhancing and simplifying the IT solution delivery experience.

The Plays…

1: Ideate solutions that meet the needs of business

The business knows its users, it owns the problem space. Apply design thinking to new product development to create high value solutions that users want to adopt.

2: Successfully navigate the project intake process

Having a clear understanding of the IM/IT project governance process will ensure your proposal can obtain the necessary approvals in order to proceed.

3: Getting started to develop your solution

Develop your team’s capacity to deliver modern digital solutions by enhancing your existing skillset through exposure to experienced people.

4: Developing a product

Methodologies and frameworks that result in working MVP solutions that have built in security and are designed for operations.

5: Transition to Ops

Following a successful initial product development and deployment a digital product is launched. At this stage a decision is made concerning the products lifecycle. Following a knowledge transfer from the development team the product can be transitioned to operations.

6: Continuous Product Improvement

In this cycle a team is responsible for a continuous cycle of product enhancement and ongoing support based on incorporating user feedback and operational metrics provided by monitoring. The product owner shifts their role to becoming the “story owner”.

7: Sustainment Lifecycle

Your product has been delivered you have a product roadmap that guides the product lifecycle.

You need to ensure you revisit your plan to ensure the annual funding is in place to account for any and all changes including product retirement and product replacement.

Definition of a Modern Application

Is an application that exhibits five important characteristics:

1. Supports multiple users through the interface of their choice
    - Web, Mobile, API (application programming interface)
2. Provides an API for accessing data over HTTP(S) for all features that would normally be available over a graphical user interface (GUI) or the command line interface (CLI)
3. Data is available via generic formats (such as JSON or XML)
4. Is built on a modern stack that supports the creation of an application with a HTTP interface and an API which can consume and emit JSON data
5. Conforms to the relevant elements of a 12-factor application

Play 1

Ideate solutions that meet the needs of business

We must begin digital projects by developing solutions for the people who will use them. Users are often but not always employees of the Ministry. Staff who interact with users on a regular basis are often an untapped source of information. Creating an environment that promotes their participation will strengthen the business unit mandate and contribute to its future success. The ideation process is less about what the product will look like, and more about who it is for, how it works and why.

Step 1: Promote innovation from within

Checklist

- Understand your business mandate.
  - Focus on business challenges. Invite your staff to contribute ideas.
  - Formulate problem statements to guide conversations.
  - Be specific, create a sense of urgency.
- Understand your user's needs.
  - Regularly review your product portfolio with users.
  - Update your feature roadmaps to reflect their needs.
  - Invite suggestions on what would enhance the way they work.
  - Can end users interact with the existing system using multiple channels? What is their experience?
- Park bias.
  - Negativity eats ideation for breakfast. Be open minded. Don't prejudge what others say.
- Encourage creativity.
  - A diverse workplace brings with it a wide range of experience. Engage an external facilitator to surface discussions.
  - Amplify soft voices. They will bring unique insights that often get overlooked. Failing to do so may lead to a disengaged workforce.
- Collaborate.
  - Time box activities. Circulate proposals. Encourage additions.
  - Require a short submission that describes objectives and benefits. Estimate effort to develop and implement. Include a risk assessment.
- Visualize.
  - A diagram helps to communicate. Develop from the end user's perspective.
- The more the better.

    - Develop as many ideas as possible around a theme. Elements of any one idea may be incorporated into another idea thereby promoting more discussion.
- Submit for Peer Review.
    - Your colleagues bring a diverse lens based on their unique insight.

Step 2: Engage your stakeholders

Checklist

- Develop & test low fidelity prototypes.
    - Describe the user's journey and interaction with the solution.
    - Develop wireframes.
    - Solicit feedback. Document attributes (goals, needs, behaviours, preferences).
- Evolve Scenarios and User Stories that speak to a set of defined Personas.
    - Invite stakeholders to review and prioritize.
    - Avoid over-solutioning how to accomplish the idea.
    - Do your research. Are your assumptions complete?

Step 3: Present Your Proposal

Checklist

- Do your homework. What would be required to go from ideation to project inception?
- Develop the product roadmap.
- Seek approval to proceed.
- Become a business priority.

Resources

Thoughtworks.com provides valuable insight into organizational future proofing in an era of constant change by promoting the concept of validated learning.

Interaction-design.org identifies strategies towards reducing potential barriers of innovation.

The BC Government Service Design Playbook provides resources, methods and templates to guide new service development. Ideao.org and Sourcemaking.com both provide additional toolkits and sources of useful information

The Canadian and US federal governments detail a series of digital plays in their playbooks. In The Government of Canada Digital Playbook 'design with users' section and the US Digital Service plays 1,2 and 3 provide useful questions to ask during ideation.

Guidance to consider

- Research - develop a fulsome understanding of the problem and the intended audience. Understand the solution context and dependencies.
- Engagement - solicit responses via surveys, workshops in order to provide data to guide the ideation process.

Play 2

Successfully navigate the Project Intake Process

To improve our ability to deliver successful IM/IT services that meet the Ministry goals and objectives, business units work with experienced senior business consultants. Following the Ministry IM/IT Governance process ensures that project proposals that have been championed by a Ministry business unit will obtain funding and approvals for strategically important initiatives.

Step 1: Obtain Business Unit endorsement

Checklist

- Follow PROSCI best practices (it all starts with change management).
- Follow your business units project authorization process.
    - Create a Concept Case/Business Case.
    - Receive approval to proceed.
    - Business Unit BA works closely with the Project Secretariat (aka: IMB) Senior Business Consultant (SBC) holds regular biweekly meetings to surface initiatives and concerns.
    - Register proposal in the Ministry project tracking system.
- Develop your operating model.
    - What does it mean to be 'done'?
    - Who will maintain and operate the solution?
    - What is the sustainment lifecycle for the product? (Skip ahead and go to Play 7 now.)

Step 2: Follow the Ministry IM-IT project intake process

Checklist

- Submit project Intake Form.
    - IM/IT Project Coordinator works with SBC to ensure submission receives visibility.
    - Provide project summary information.
    - Complete client project survey in order to rank submission.
- Provide resources to participate at the Architecture Review Board(ARB).
    - ARB assesses submission to determine technical fit with GCIO/MCIO direction.
    - Project review by the Project Review Board (PRAB).

- Upon approval to proceed a project charter will be required.
- Project submission will be made to the OCIO-DIO and authorized by the DMC.

Step 3: Determine Project Resourcing

Checklist

- Internally managed initiative.
  - Business Unit has the skillset and resources to successfully manage and deliver the project.
  - Business Unit enters a procurement activity to obtain an IM/IT solution (RFP).
- Partnership Agreement.
  - Business Unit enters into an Alliance agreement with the BCDevExchange/OCIO-DPDD to design and develop a solution -*or*-
  - Business Unit enters a "Memorandum of Understanding" (MOU) with the IMB to co-develop and deliver an IM/IT solution.
    - Incorporate a transition plan section into your MOU.
    - Formulate a common understanding of the 'definition of done'.
  - **NOTE:** It takes time to resource additional skills, often as long as 6 months, plan accordingly in your transition plan -*or*-
  - Draw down on Ministry contract for an external vendor (such as CGI) via the IMBs Request for Service (RFS) process.

Step 4: Provide regular updates

Checklist

- Based upon the reporting requirement.
  - IM/IT capital = quarterly.
  - Project reporting = monthly.
- Working with the Secretariat Portfolio Manager / Project Analyst provide regular status update to Ministry executives.

Resources

The CITZ IMB maintains a collection of templates and information in their Project Management Resource Centre. Gaurav Mehra provides insight into common challenges encountered in Agile intake processes: Part 1 & Part 2

Guidance to consider

Broad examples include:

- Understand the Work Intake Process.
- Architecture Review Board should be aware of your project and may ask you to provide a presentation.
- Understand why Governance is important.
- Adhere to Government IM/IT Principles and Standards.

Play 3

Getting Started to Develop Your Solution

Project success is never guaranteed. Determine your approach to product development in order to correctly fit your resource needs. Develop your team's capacity to deliver modern digital solutions by enhancing your existing skillset through exposure to experienced people either using temporary assignments, contractors or co-ops. Once you have your team come to a common understanding of the culture you wish to promote then entrust design and development to the team which you have empowered.

Have a design system which establishes your product framework based on your priorities and complies with government standards.

Step 1: Go Agile, don't repeat the old ways

Checklist

- Select Agile because:
  - The intended solution is large.
  - The solution can be a collection of many different projects.
  - Working within a Time & Materials contract.
  - Requirements are expected to change over time.
  - Product Owners are embedded in a team and prepare requirements daily.
  - Projects do not have a Project Manager as all team members co-manage the project.
  - Developers have cross-functional skills.
- Reject the 'old ways' (aka: Waterfall) because:
  - The intended solution will be large and has a higher risk for failure.
  - Requirements are expected to change over time.
  - Many waterfall projects are abandoned and do not meet user expectations when finally completed.

Step 2: Augment your team with experience

Checklist

- Identify the roles your project needs.
  - Build up your team's capacity by bringing in resources that are seasoned professionals.
  - Work with the procurement team to understand how to evaluate proposals.

    - Provide existing staff with opportunities to pair up with contractors to build more internal capability.
    - Identify the product/project owner.
    - Include a Security and Privacy Officer in your team.
- Consider the Sprint with Us and Code With Us mechanisms that allows the Government of B.C. to procure Agile software development teams.

Step 3: Meet early and follow up often

Checklist

- Review the project team membership identify responsibilities.
- Specify the approval process for signing off deliverables.
- Review the Project Charter/Statement of Work to ensure common understanding of purpose, objectives and success criteria.
- Discuss the client's attitude and approach to managing risk and change.
- Agree on a reporting strategy to track and communicate progress to stakeholders.
- Agree on which tools will foster the most productive collaboration.
- What assets are required in order to start delivering products.
- Formulate the Project Inception Agenda.

Step 4: Project Inception

Checklist

- Identify the Scrum Master and Product Owner.
- Have a team agreement session, identify the team's values. Display prominently.
- Establish "way of working" (sprint cycles, way of communication, meeting regularity).
- Agree on a common "definition of done".
- Create virtual/physical workspace for the team.
- Set up your development environment.
- Define initial backlog of requirements/stories/features.
- Identify the high value user stories based on their needs/challenges formulated as user stories (Waterfall). Specify acceptance criteria for each story.
- Specify acceptance criteria for each story.

Step 5: Prepare to Sprint

Checklist

- Review your Definition of Ready.

- Publish your Sprint calendar, identify 'Epics' ,key project dates. Team resource vacations.
- Advertise when you will hold Sprint Planning, Stand-ups, Sprint Reviews and Product/feature Demos.
- Work in the open. Deliver often. Increment continuously.

Resources

Adopt a set of Digital Principles and Standards such as these. The Government of Canada Digital Playbook has a useful tool to help determine if 'Agile' is right for your project. For more information on the BCDevExchange approach to agile consult the DevHub. The US Digital Playbook describes the benefits of adopting an incremental agile approach to digital project development.

Tooling to consider

Broad examples include:

- Kanban board
- Task assignment system (Jira)
- Documentation Hub (Confluence)
- Code Repository (GitHub)
- Create a training schedule
- Scrum templates
- Adopt Scrum ceremonies

Play 4

Develop your Product

Adopt code frameworks that accelerate your development to create working Minimal Viable Product (MVP) solutions which have built in security and are designed for product management.

Define your architecturally significant requirements (ASR) based on product "abilities":

- Deployability
- Modifiability
- Testability
- Monitorability

Step 1: Meet with your team

- Establish a team agreement.

Step 2: Design / describe your Solution (conceptual architecture)

- Whiteboarding – relationships between components, data model
- Skeletal framework – overarching view of the system (placement of function technical ex: use ArchiMate to visualize relationships)

Step 3: Adopt the 12 factor approach

Step 4: Understand 'cloud native architecture'

Step 5: Set up your environment (technical infrastructure)

What are the prescriptive patterns to get you started?

- Access to development environments
- Communication channels for support (e.g.: MS-Teams, RocketChat, Slack)

Step 6: Identify your development workflow

- Development processes

Step 7: Set up, document & implement your tooling (code repository, pipelines, automation)

Verify your processes for successful:

- Integration
- Build and Deployment
- Application Security

Step 8: Start Coding, testing and commit to your source code management system

- (Where it makes sense…) adopt a Test Driven Development (TDD) approach to coding.

Step 9: Submit for Code Reviews (as required)

Step 10: Complete User Acceptance testing and remediate defects

Step 11: Deploy, promote through your environments

- Product Owner (PO) confirms the Definition of Done (DoD) prior to deployment.

Step 12: Demonstrate your products frequently, obtain user feedback

- Sprint demos (feature delivery)
- Stakeholder demo (batch delivery)

Step 13: Monitor and measure the development process

- Supports Product Owner , supports the Team – shields members from scope/feature creep
- Monitoring your system for reliability

Definition of an MVP:

A Minimal Viable Product (MVP) is one that delivers core functionality as expected by a business units' target audience. The core functionality is a set of features that is delivered at the initial product launch. Once launched the product follows a documented lifecycle for user feedback and product support.

An MVP is no longer considered to be an experiment, but a product that has value to its users.

https://en.wikipedia.org/wiki/Minimum_viable_product

Guidance to consider

- Understand Agile and Software Product Integrity.
- Define and develop your Pipeline (ex: Jenkins).
- Create and maintain a Test Harness.

Play 5

Transition to Ops

Following a successful initial product development and deployment a new modern application (digital product) is launched. At this stage a decision is made concerning the products lifecycle. Ensure that the Operations team has been involved in the previous plays on a regular basis from team formation product delivery so that they are aware and have been able to provide input into your delivery pipeline processes. Following formal knowledge transfer from the development team the product can be transitioned to the operations team. If the initial product (often called a Minimal Viable Product or MVP) will require enhancements, then it typically enters the **Continuous Product Improvement** cycle as described in Play 6. If however, the product meets the business needs, it will proceed through an operations cycle consisting of product governance and change management.

Step 1: Review your operations MOU (that was part of Play 2 with the Ministry IT Operations branch (or outsourced provider)

- It takes time to resource additional skills, often as long as 6 months, plan accordingly in your transition plan

Step 2: Complete knowledge transfer

- adopt the 'Site Reliability Engineer' support model pioneered by Google
- throughout the development cycle the product team member who represents the Ministry Ops team should be regularly participating in the development team meetings and ideally is cross functionally trained to understand how product features have been developed, tested and pushed through the deployment pipeline

Step 3: Validate Ops processes including:

- Change Management procedures
- Monitoring & Logging
    - Application Performance Monitoring (APM) metrics have been defined
    - Application logs are generated and maintained in accordance with policy.
- Tooling Updates
- Product Triage
    - Define your product triage procedures
- Escalation

Guidance to consider:

- The Ops team member should communicate with colleagues of the Ministry Ops team to educate other team members on the product shared documentation, its tooling and design review meetings
- Source Code Repository (such as the projects GitHub repo) has a readme file with relevant procedures and links required to recreate the solution installation
- Make documentation easily locatable (Confluence, SharePoint, GitHub)
- Create a Product "Runbook"
- Monitoring Environment (ELK stack, Splunk, Nagios)
- Google's monitoring & logging patterns
- Triage best practices
- Knowledge Transfer best practices
- Review your operating commitments
    - 9's uptime
    - Performance targets

Play 6

Continuous Product Improvement

In this cycle a team is responsible for a continuous cycle of product enhancement and ongoing support based on incorporating user feedback and operational metrics provided by monitoring. The product owner shifts their role to becoming the “story owner”.

Ensure that team ownership (that was originally defined in Play 2) has been reviewed and reaffirmed.

- Is it within the product business unit?
- Has it been delegated to the IMB (through a MOU agreement)?
- Has it been outsourced to an independent services organization?
- Has an Application/Product Manager role been defined?
- Who is providing the Business Analyst (BA) services? Are the IMB BA’s involved?

Step 1: Kanban drives the developers [who add features]

Step 2: Incorporate Feedback

Step 3: Prioritize bugs and new feature requests (based on the PO’s decision)

Step 4: Ongoing Code / Build / Deploy / Release Management

Step 5: Iteratively improve your processes

- Seek to improve the sizing of tasks.
- Strive for better team and stakeholder collaboration.

Step 6: Skills retention requires a plan

- Change is inevitable, ensure you have a succession plan in place for each ‘role’ on your team .
- Review your processes, review your documentation.
- Share knowledge .
- Ask for feedback from other groups.
- Seek opportunities to invite other interested workers into the team (via ‘Expression of Interest’ Temporary Assignments).
- Engage University/College Co-op students.
- Give staff the time to take webinars and other online training.

Step 7: Communicate your success

- Measure, celebrate and communicate the team's workflow throughout the organization through announcements such as newsletters, provide 'lessons learned' webinars and presentations and author blogs.

Guidance to consider:

- Does your team composition (IS21, IS24, IS27s) still reflect the products' needs?
- Does your team have access to Service Design and User Experience (UX) skills as needed?
- Kanban board guides activities.
- Incident/Bug Tickets can be assigned.
- Task assignment system (Jira) is being regularly reviewed.

Play 7

Sustainment Lifecycle

Your product has been delivered and you have a product roadmap that guides the product lifecycle.

Maintain care and feeding of your processes and your stakeholders through good governance.

You need to ensure you revisit your plans to ensure that annual funding is in place to account for any and all changes including product retirement and product replacement.

Step 1: Maintain Happy User

- Review your products goals and assumptions - are they still relevant?
- Evaluate and Analyze product analytics data (if available, if not look to create it).
- Interpret user feedback.
- Conduct product research.

Step 2: Maintain product funding

- Review your products' value proposition.
- Get ahead of the budget cycle, review your opex regularly, update your budget plan often.
- Understand opportunities to adopt nonproprietary services and products.
- Seek reviews from the finance team.

Step 3: Maintain vendor / partner relationships (including your support contracts)

Step 4: Continuously improve your overall lifecycle

- The goal is continuous productivity improvement.

Step 5: Maintain product integrity & quality

- Product Owner and/or the Application Manager's responsibility

Guidance to consider:

- Business Unit Budgeting Exercise
- User Engagement
- UX review
- Industry Analysis
- Review the corporate product inventory (look for new opportunities)
- US Department of Defense LCSP (as an example)

CITZ IMB MODERN APPLICATION PLAYBOOK CHECKLIST

Product:		
Owner:		
Date:		
Notes:		
Guidance:	*Where followed briefly describe measures that where undertaken to adhere to the intent of the "play", otherwise state why the step was not followed.*	
Play 1: Ideate solutions that meet the needs of business		
Step 1:	Promote innovation from within	
Step 2:	Engage your stakeholders	
Step 3:	Present your proposal	
Play 2: Successfully navigate the Project intake Process		
Step 1:	Obtain Business Unit endorsement	
Step 2:	Follow the Ministry IM-IT project intake process	
Step 3:	Determine Project Resourcing	
Step 4:	Provide regular updates	
Play 3: Getting Started to Develop Your Solution		
Step 1:	Go Agile, don't repeat the old ways	

Step 2:	Augment your team with experience	
Step 3:	Meet early and follow up often	
Step 4:	Project inception	
Step 5:	Prepare to Sprint	
Play 4: Develop your Product		
Step 1:	Meet with your team	
Step 2:	Design /describe your Solution (conceptual architecture)	
Step 3:	Adopt the 12 factor approach	
Step 4:	Understand ‘cloud native architecture’	
Step 5:	Setup your environment (technical infrastructure)	
Step 6:	Identify your development workflow	
Step 7:	Set up, document & implement your tooling	
Step 8:	Start coding, testing and commit to your source code management system	
Step 9:	Submit for Code Reviews	
Step 10:	Complete user Acceptance testing and remediate defects	

Step 11:	Deploy, promote through your environments	
Step 12:	Demonstrate your product frequently, obtain user feedback	
Step 13:	Monitor and measure the development process	
Play 5: Transition to Ops		
Step 1:	Review your operations MOU	
Step 2:	Complete knowledge transfer	
Step 3:	Validate Ops processes	
Play 6: Continuous Product Improvement		
Step 1:	Kanban drives the developers	
Step 2:	Incorporate Feedback	
Step 3:	Prioritize bugs and new feature requests	
Step 4:	Ongoing Code/Build/Deploy/Release Management	
Step 5:	Iteratively improve your processes	
Step 6:	Skills retention requires a plan	
Step 7:	Communicate your success	

Play 7: Sustainment Lifecycle		
Step 1:	Maintain Happy User	
Step 2:	Maintain product funding	
Step 3:	Maintain vendor/partner relationships	
Step 4:	Continuously improve your overall lifecycle	
Step 5:	Maintain product integrity & quality	

Modern Application Development
IMB's Principles

Adopting these principles can help to guide your product decisions. Designing and building solutions that are loosely coupled and utilize interfaces for process communication will result in maintainable applications.

The following **proposed** principles offer guiding statements that should be considered in the development of your applications.

- Guiding Principles
- Application Principles
- Architecture Principles
- Project Governance Principles
- Technical Principles
- Infrastructure Principles

IMB is developing a set of guidance patterns to assist Ministry business units in selecting platforms for their line of business solutions as well as a set of engagement patterns for IMB services. This list will evolve over time.

Platform & Solution Patterns

- DevExchange Adoption Pattern
- Microsoft PowerBI Adoption Pattern
- Microsoft Dynamics Adoption Pattern
- Microsoft SharePoint Adoption Pattern
- RHEL OpenShift Adoption Pattern
- Web Application Development Pattern

IMB Patterns

- Ministry IMB Business Unit Engagement Pattern
- Ministry IMB Service Desk Pattern
- Ministry STRA & PIA Pattern
- Ministry IMB Infrastructure Support Pattern

Guiding Principles *adopted*

Principle 1: Develop open and innovative partnerships

We recognize that by developing trusted partnerships and collaborating across teams and the ministry, we create opportunities to find new ways of delivering efficient and effective services to program areas.

Principle 2: Service focus

Solutions and services are designed from a client-centered and end-to-end digital service delivery perspective to increase the value they bring to the client.

Principle 3: Teamwork and collaboration

We will create and empower cross-functional, dynamic teams to increase engagement, communication, talent and opportunities to deliver value and services to clients.

Principle 4: Invest wisely

Through effective governance and financial oversight, we will maximize IM/IT spend and optimize our existing investments to support the ministry.

Principle 5: Enable a modern and innovative workplace

We will support collaboration, create efficiencies, and encourage people to work smarter, greener, healthier and more innovatively so all feel valued and recognized.

Principle 6: Continuous improvement

We believe in testing early and often. We will test processes, services, and technology end-to-end and continuously improve in response to user feedback.

Principle 7: Proactive approach to security and privacy

We are committed to supporting our clients to adopt a proactive approach to ensuring strong information security and privacy protection practices. By applying a 'privacy and security by design' philosophy, we help our clients ensure that privacy and security measures are considered at project initiation and built into the solution, rather than having to be retrofitted afterwards.

Application Principles *proposed*

Principle 1: Do no harm

Only build what you can't buy and buy what you can't build.

Principle 2: Build/Buy responsibly

Don't contribute to technical debt.

Principle 3: Be citizen-centric

Everything you build is for the greater good. Citizens are the real product owners.

Principle 4: Today's build could be tomorrows burden

Like building responsibly, think about the effort required to maintain and evolve the application.

Principle 5: Nature vs. Nurture

An application will only be as good as the effort put into its evolution.

Principle 6: Sharing is caring

Information is a commodity to be shared; build.

Principle 7: Keep it Small

Small code packages, small feature sets make for shorter delivery cycles, fewer changes and overall better software quality.

Principle 8: Focus on the developer

Select the best environment with the right tooling. Endorse a good set of use of Architecture Principles.

Principle 9: Develop for Networks

Application communication happens over the network, not in memory. Supports distributed development teams, increase application resiliency, and simplify product deployment.

Architecture Principles *proposed*

Principle 1: Primacy of Principles

… TOGAF Principles, OCIO Digital principles, HADF principles

Principle 2: Adhere to Standards

… OCIO Digital Framework, GCIO and Ministry Standards

Principle 3: Adopt a set of Software Design Principles

Patterns and practices are the tools used to achieve the desired outcome of the principles. Follow fundamental principles for writing quality software such as:

- KISS — Keep it simple, stupid.
- DRY—Don't repeat yourself.
    - Do not duplicate application this is a frequent source of errors.
- YAGNI—You aren't gonna need it.
    - Are you going to need that feature? If not: leave it out.
- SoC—Separation of concerns.
    - Minimize tight coupling of code to low-implementation details by separating core business logic from infrastructure and user interface logic so that the module is easy to test and can be evolved.

Principle 4: Adopt a Service Oriented Architecture

- Single responsibility
    - Objects should have only one reason to change this helps to produce more loosely coupled and modular systems.
- Bounded Context
    - Reduce complexity by reducing a solution into separate conceptual modules where each module represents a context that is separated from other contexts (i.e.: bounded) and can be evolved independently of one another.

Principle 5: API First

Develop your solution to be used by multiple client applications through a well described API.

Principle 6: Be data driven

Principle 7: Design to be secure

Design, develop and deliver solutions that mitigate risks. Ensure security is addressed end to end and considered upfront.

Principle 8: Adopt Cloud

Whether public, private or hybrid cloud, adopt a set of guiding principles for Cloud Computing and Use - include enablement, cost/benefit, enterprise risk, capability, accountability and trust.

Principle 9: DevOps for Agility

DevOps is founded on product delivery. Agile is founded on the project success. Embrace the combination of 1) continuous integration (including Build management, test management and automation), 2) continuous delivery (including environment management and deployment management), 3) infrastructure as code and 4) iterative development approach to support successful projects.

Project Governance Principles *proposed*

These principles apply to the **ministry governance process for the planning, intake and approval of IM/IT projects**.

Goals of Principles

- These principles will help the project governance team ensure that we understand the scope of the work and stay focused on outcomes.
- The principles should be simple, straightforward and high-level.

Principle 1: Enable the right people to make the right decisions at the right times.

Principle 2: Be easy to understand and follow.

Principle 3: Be simple and efficient, with each stepping adding value.

Principle 4: Be timely.

Principle 5: Add value for all stakeholders, while maximizing the benefits to the Ministry.

Principle 6: Be transparent and enable decisions that are objective and fair.

Principle 7: Apply a common set of selection criteria to all proposed IM/IT projects.

Technical Principles *proposed*

Principle 1: Prioritize Principles

Any decision to ignore or reject core principles so as to satisfy project goals must be recorded.

Principle 2: Adopt Technology Standards that aid the Business

- Make Friends in the Business – if the business understands why a standard need to be adopted, they will be more likely to weigh the implications on impacts to their expected outcomes.
- Stakeholders drive the adoption of Technical Standards – if a standard is seen as an impediment then the natural pattern is to seek an exemption. Ensuring stakeholders are aware early in the project lifecycle helps to develop adoption strategies.
- Involve the Architects from start of your project – Architects are central to product viability discussions. Engaging them early will mitigate waste in the product development flow.

Infrastructure Principles *proposed*

Principle 1: Plan for needed Capacity

Solutions that evolve requirements need to be scalable in order to guarantee performance over the entire product lifecycle. Don't over commit up front, validate the ability to meet unexpected system demands.

Principle 2: Design for Scalability

Adoption and feature growth will result in unforeseen system demands. Ensure that the solution architecture is able to scale.

Principle 3: Achieve User Happiness through System Performance

User expectations will change over time. Ensure that the architecture is adaptable and can satisfy performance demands. Assess how the solution is hosted and its portability.

Principle 4: Monitoring enables your team

Understand what to monitor. Ensure that the available monitoring system can provide actionable metrics in order to provide the necessary information needed to resolve any issues that can arise. Knowing what can cause performance issues and monitoring for those scenarios will feed the product continuous lifecycle with data.

Modern Application Development IMB Patterns and Practices

Patterns describe typical issues that are experienced by Ministry business clients when they start to develop a new IM/IT based solution. A pattern provides techniques that describe best practices toward achieving a desired outcome.

When a Ministry business unit starts its product development journey it is often aware of the need to get a STRA and if required, a PIA. Ministry clients require additional information when starting their new product journeys. The IMB is compiling a list of repeatable adoption and usage "patterns" to common questions asked of our Ministry business consultants. The following list of patterns will evolve as IMB works with our clients to develop additional modern solutions based on our supported platforms and service offerings

How are patterns described?

A pattern describes a process and a 'thing'. It describes a proven solution that can be applied by our clients. Patterns are usually generic and intended to initiate a dialog.

A pattern has a:

- name & short summary
  - which demonstrates the business need for the pattern
- context
  - situations in which the pattern may apply
- scenario
  - that the pattern addresses
- solution
  - the fundamental solution concept that addresses the scenario

IMB Patterns we will evolve will include…

Platform & Solution Patterns

- DevExchange Adoption Pattern
  - Describes a set of steps to allow your team to start a DevOps journey in the BC Dev Exchange
- Microsoft PowerBI Adoption Pattern
  - Describes a set of steps to allow your team to start a journey on the Microsoft PowerBI platform.
- Microsoft Dynamics Adoption Pattern
  - Describes a set of steps to allow your team to start a journey on the Microsoft Dynamics platform.
- Microsoft SharePoint Adoption Pattern
  - Describes a set of steps to allow your team to start a journey on the Microsoft SharePoint platform.
- RHEL OpenShift Adoption Pattern
  - Describes a set of steps to allow your team to start a journey on the OpenShift platform.
- Web Application Development Pattern
  - *currently in development*
  - Describes a set of steps to allow your team to start a journey when developing a progressive/ responsive web application.

IMB Patterns

- Ministry IMB Business Unit Engagement Pattern
  - Describes a set of steps to allow your team to start a journey with the IMB.
- Ministry IMB Service Desk Pattern
  - Describes a set of steps to follow when working with the IMB Service Desk.
- Ministry STRA & PIA Pattern
  - Describes a set of steps to allow your team to start your STRA journey.
- Ministry IMB Technical Services Support Pattern
  - Describes a set of services available to your team and how to engage the IMB Technical Services team.

Ministry STRA & PIA Pattern

Describes a set of steps to allow your team to start your STRA and PIA journey.

- Project team contacts the Information Privacy and Security Team (IPS) to seek advice and initiate the assessment process.
- IPS collects information from the project team to provide an initial recommendation regarding assessment approach and scope.
- IPS works together with the project team, and other key stakeholders as needed, to conduct the assessment.
- Project team supports the process by providing all relevant information and documentation to IPS in order to conduct the assessment.
- IPS liaises with core government information privacy and security branches (i.e., the Privacy, Compliance and Training Branch (PCT) and the Information Security Branch (ISB)) throughout the process, including during the assessment 'sign-off' phase.

Ministry IMB Technical Services Support Pattern

Describes a set of steps to allow your team to engage the IMB Infrastructure support team and their service offerings.

IMB Technical Services works with the different business units to manage their technical infrastructure.

- Project team contacts the IMB Technical Services team to seek advice on:
  - Management and provisioning of Infrastructure.
  - SSL cert management
  - Assistance with Reverse Proxy and Site Minder setup for applications and websites.
  - Monitoring Server and application infrastructure.
  - Assistance with setup, install and upgrades for applications and Web infrastructure to support applications.
  - Firewall rule management.
  - Infrastructure patch and change management.
- IMB Technical Services Team is available to provide consulting services on these services.

Glossary of Useful Links

#
12 factor
A
Acceptance Criteria
Agile
Application Manager
Application Performance (APM)
Architecture Review Board (ARB)
Architecture significant requirements (ASR)
B
backlog
BC Dev Exchange
Build
Business Case
Business Analyst
C
change management
Citizen Services (CITZ)
cloud native
Code review
Code repository
Code With Us (CWU)
Concept Case
D
Definition of Done (DoD)
Definition of Ready (DOR)
Design System
Design Thinking
Deploy
Devhub
DevOps
Digital framework
Digital Playbook (Canada)
Digital Playbook (USA)
E
ELK stack
EPICS
escalation
F
Feedback
Frameworks
G
GitHub
H

I

Information Management Branch (IMB)

J

Jenkins

K

Kanban

Knowledge Transfer

L

Lifecycle

Logging

Low Fidelity prototype

M

Methodologies (Agile)

Ministry Chief information Officer (MCIO)

MERN Stack

Ministry Information Security Officer (MISO)

Minimal Viable Product (MVP)

Metrics (application performance)

Modern Application

Monitoring

Memorandum of Understanding (MOU)

N

O

Office of the Chief Information officer (OCIO)

OCIO Digital Investment Office (OCIO-DIO)

Digital Investment office of the BC OCIO

Operations (Ops)

Operating Expense (OpEx)

P

Persona

Pipeline (agile)

Playbooks (examples)

Principles (example)

Product Owner (PO)

Progressive Web App (PWA)

Project Charter

Project Inception

Project Governance

Project Review Board

Q

R

Release Management

Risk Assessment (IT)

Roadmap

Runbook

S

Scenarios
Scrum Master
Security
Senior Business Consultant
Single Page Application (SPA)
Site Reliability Engineer (SRE)
Sprints
Sprint demo
Sprint retrospective
Sprint with Us (SWU)
Statement of Work (SOW)
Steering Committee
Story Owner

T

Test Driven Development (TDD)
triage

U

User Experience (UX)
User Interface (UI)
User Journey Mapping
User Story

V

Value proposition

W

Way of working (WOW)
waterfall
wireframes

X

Y

Z

Working with the BC Web Design Jekyll Theme

This GitHub Pages theme is based on the 18F + US Web Design System documentation template described here.

The 18F source GitHub is available at this link where you will find information on how to apply the base theme including configuration for your own use.

In our clone we have replaced all references from ‘uswds’ to ‘bcwds’.

Additional customizations include the BC Government masthead, header and footer elements.

The BCSans typography has also been incorporated.

This work is made available under the BC DevExchange Apache 2.0 license with no warranty implied or expressed.

The Playbook Github repo is available at https://github.com/bcgov/CITZ-IMB-playbook