Agile Terrain Development for Simulation-Based Training

Rick Osborne
The MITRE Corporation
Orlando, FL
rosborne@mitre.org

Donald Washburn
Leidos
Orlando, FL
donald.a.washburn@leidos.com

Mark Faulk
Cornerstone Software Solutions, Inc.
Orlando, FL
mfaulk@cssflorida.net

Ronald G. Moore
Leidos
Orlando, FL
ronald.g.moore@leidos.com

ABSTRACT

A large portion of the global population lives in areas of extraordinary geographic size and population density commonly referred to as megacities. Maintaining social stability and ensuring citizen safety in these megacities is a formidable challenge for both civilian and military forces. Disruptive technologies, such as mobile communication devices and social media, can result in near-instant social volatility. Expanded situational awareness, continuous real-time monitoring, and improved reaction and response methods are essential.

Several U.S. Department of Defense (DoD) organizations are working to define how military operations will address megacity-related challenges and what is required to train for these operations. It is unreasonable to define all the megacity geospatial requirements upfront for simulation-based training. Megacities include manmade and natural structures (surface, subsurface, and super-surface); utility networks; complex building interiors; patterns-of-life and emergency response data; sophisticated building usage information; and detailed transportation networks.

This paper presents an agile terrain development process to support simulation-based training. Agile frameworks support iterative product development. When enhancing a product iteratively, the product team does not start with a full specification of requirements. Instead, development begins by specifying only the well-known requirements and implementing just part of the product, which can then be reviewed to identify further requirements. The team repeats this process, producing a new version of the product that builds upon prior versions. The agile terrain development process assumes there is a terrain data repository that consists of low-fidelity geospatial data for the world and higher-fidelity data for some areas. By leveraging this process, features can be iteratively added to the repository to meet training requirements, reduce cost, and reduce the time to deliver terrain databases to the warfighter. The paper also applies the process to two training examples to demonstrate its utility.

ABOUT THE AUTHORS

Mr. Rick Osborne is a Lead Simulation Engineer at The MITRE Corporation. He is currently supporting the Synthetic Environment Core (SE Core) program at U.S. Army PEO STRI. His current interests include microservice architectures, game development, and DevOps. He earned his B.S. in Computer Engineering from Christopher Newport University, M.E. in Computer Engineering from Old Dominion University, and M.S. in Modeling and Simulation from University of Central Florida.

Mr. Donald A. Washburn is a Senior Systems Engineer at Leidos in Orlando, FL. He is currently serving as engineering deputy lead on the SE Core Common Virtual Environment Management (CVEM) program. Don has vast leadership experience in all aspects of project management, excelling at planning and customer interfacing. He successfully managed multi-functional teams responsible for developing hardware, software, and network products through all phases of project development from proposal to maintenance. Mr. Washburn received his B.S. and M.S. in Industrial Engineering from the University of Central Florida. He has been working in the simulation and training industry for more than 25 years with extensive experience in software, systems, and leadership roles.

Mr. Mark A. Faulk is a Systems Engineer at Cornerstone Software Solutions, Inc. in Orlando, FL. He is currently the Lead Systems Engineer for the Army SE Core CVEM program. Mr. Faulk has over 30 years of experience in large systems development and over 20 years of experience with simulation and training systems architectures, systems engineering, software development, terrain generation, and interoperability.

Mr. Ronald Moore is currently the Chief Architect for SE Core CVEM. He has over 35 years of experience in the modeling, simulation and training industry with expertise in software development, computer graphics, computer image generation, simulation geospatial terrain database production, sound simulation, streaming audio and video, and PC and console game development.

Agile Terrain Development for Simulation-Based Training

Rick Osborne
The MITRE Corporation
Orlando, FL
rosborne@mitre.org

Donald Washburn
Leidos
Orlando, FL
donald.a.washburn@leidos.com

Mark Faulk
Cornerstone Software Solutions, Inc.
Orlando, FL
mfaulk@cssflorida.net

Ronald G. Moore
Leidos
Orlando, FL
ronald.g.moore@leidos.com

INTRODUCTION

With increased populations living in megacities worldwide, simulation-based training in dense urban environments has become both a priority and a challenge. The scale and complexity as well as the evolving human and technology landscape are challenging traditional approaches to urban warfare and civilian operations doctrine and training. Disruptive technologies, such as mobile communications and social media, bring near instant environment changes requiring entirely new situational awareness and reaction training.

Several U.S. Department of Defense (DoD) organizations are working to define how military operations will address megacity-related challenges and what is required to train for these operations. According to a U.S. Army TRADOC G-2 report concerning megacities, the types of tasks that the Army may be required to perform in a megacity environment include:

> Non-combatant evacuation; humanitarian assistance disaster relief (HADR) missions; raids; deny adversary objectives; counter weapons of mass destruction operations; conduct military engagement and security cooperation; provide a global stabilizing presence; provide support to civil authorities, and counter terrorism/counterinsurgency missions (Farrel & Ward, 2016).

Other U.S. military services and coalition partners will undoubtedly need to perform similar tasks.

Megacities include manmade and natural structures (surface, subsurface, and super-surface); utility networks; complex building interiors; patterns-of-life and emergency response data; sophisticated buildings usage information; and detailed transportation networks. Both training and geospatial requirements will continuously evolve as service personnel learn more about the operational environment. Therefore, it is unreasonable to define all the training system and geospatial requirements for megacities up front.

To accommodate evolving requirements, this paper presents an approach for agile terrain development. Agile frameworks and methodologies support iterative product development. When developing a product iteratively, the product team does not need to identify the full set of requirements at the start. Instead, development can begin with a minimal set of requirements, typically those already known, and implementing the top priority requirements first. As the product evolves through iterations, subsequent requirements can be added to the list for future implementation. The proposed process assumes there is already a terrain data repository that consists of low-fidelity geospatial data for the entire world and higher-fidelity data in some areas. By leveraging this process, the developer can iteratively add features to this repository to satisfy training requirements, reduce cost, and reduce the time to deliver terrain databases to the warfighter.

This paper also steps the reader through the process using two scenarios to demonstrate its utility. Both scenarios are taken and adapted from the *Department of Homeland Security National Planning Scenarios: Executive Summaries* publication (Howe, 2004). To frame our understanding of what megacities are, we focused on Urban Operations as described in *Joint Publication 3-06* (Joint Publication 3-06: Joint Urban Operations, 2013) and *US Army Field Manual 3-06* (FM 3-06: Urban Operations, 2006).

- Chapter 4 of FM 3-06 describes how megacities affect Warfighting Functions and Tactics in the areas of: Intelligence, Movement and Maneuver, Fire Support, Protection, Sustainment, and Command and Control.

- The concept of Intelligent Preparation of the Battlefield (IPB) is a significant focus in FM 3-06. Central to the IPB process, which is performed continuously at all levels, is the identification of the terrain, societal, and infrastructure characteristics of the urban environment. Examination of Significant Urban Characteristics lists found in Appendix B of FM 3-06 provided the main source of assessing which features or data the megacities should incorporate.

BACKGROUND

The U.S. Army Integrated Training Environment (ITE) terrain database development process, supported by the Synthetic Environment Core (SE Core) program, is composed of the following steps: definition of the training play-box and content requirements, building the database, testing the correlated runtime formats, and delivering the database to the ITE systems maintainers for fielding. The ITE terrain database generation process follows a traditional waterfall development process, which limits the content of the terrain databases, and subsequently the functionality of the Army training systems which use it. Within the last few years, SE Core has begun supporting the creation of specialty Games-For-Training (GFT) databases, derived from ITE Master Database (MDB) content, to satisfy near-term, unique training requirements. For example, SE Core was able to quickly build an active shooter response terrain database leveraging an ITE database, by adding seven unique enterable buildings to the MDB data. This resulted in a one-off GFT specialty database. Specialty database requests such as this demonstrate the need for a more agile approach to database generation in order to be responsive to constantly evolving training needs.

Figure 1 Active Shooter Response - Enterable Building

AGILE TERRAIN DEVELOPMENT PROCESS

In this section, we discuss our agile terrain development process, which is based on Scrum (Sutherland & Schwaber, 1995). Scrum is a widely used framework for iterative product development. The Scrum Framework includes roles, events, artifacts, and rules that bind them together. Our proposed Scrum implementation for agile terrain development describes the roles, events, artifacts, and rules as they relate to terrain database development. This section contains two parts. The first part is an overview of the roles and events with the artifacts and rules discussed within the context of a specific role or event. The second part provides standard steps Terrain Developers would perform throughout the process. Although this paper provides a summary of Scrum, it does not go into great detail. For further information, please see the authoritative source on Scrum, *The Scrum Guide* (Schwaber & Sutherland, 2017).

Roles - Scrum roles include: *Product Owner, Scrum Master*, and *Development Team*. The Product Owner is a project's key stakeholder who provides the vision of what to build and conveys that vision to the Scrum Team. The Product Owner is also the person responsible for managing the *Product Backlog*, which is an ordered list of the product's requirements. For terrain database development, the Product Owner is a representative of the training community who is responsible for managing requirements, tradeoffs, and coordination between the acquisition and user communities. The Development Team is a cross-functional group of professionals who have the necessary expertise to create the product. In this context, the Development Team should include both Simulation Developers as well as a Terrain Developers. The Simulation Developers know what training tasks can be exercised within the simulation, know what terrain data the simulation consumes, and know whether the Simulation Team needs to modify the simulation to support a given training requirement. The Terrain Developers know how to identify existing geospatial terrain data, to leverage new sources of geographic information system (GIS) data, to conflate the GIS data, to generate data when data within an area of interest is not available, and to transform the data into the needed runtime formats. Additionally, the Simulation Developers and Terrain Developers are responsible for identifying tradeoffs to the Product Owner when their work capacity has been reached. The Scrum Master is responsible for promoting and supporting the Scrum framework by helping everyone understand its theory, practices, rules, and values. A Scrum Master is also responsible for removing impediments that are preventing the Development Team from making progress on the product.

Events **-** At the core of the Scrum process is the *Sprint*. A Sprint includes *Sprint Planning*, *Development*, *Daily Scrum*, *Sprint Review*, and *Sprint Retrospective* as shown in Figure 2. It has a time-box of one month or less during which the Development Team produces a potentially releasable *product increment*. In our case, the product increment is a terrain database. Figure 2 depicts a series of Scrum events, and the outputs of those events (Scrum artifacts). An additional event, *Backlog Refinement*, which is not an official Scrum event but is considered a Scrum best practice is included in our implementation.

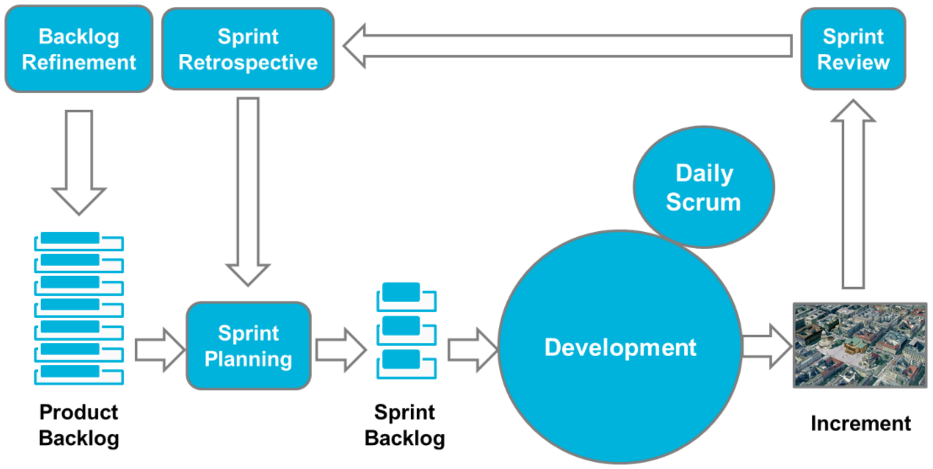

Figure 2 Typical Scrum Sprint

Before the Scrum Team can start a Sprint, it must have a populated Product Backlog. The Scrum Team will likely have several Backlog Refinement events before beginning the first Sprint. Backlog Refinement is the act of adding detail, estimates, and order to requirements in the Product Backlog. The Scrum Team adds new requirements and refines existing requirements during this event. The output of this event is a groomed Product Backlog that includes prioritized and well-defined requirements. For terrain development, we assume the Product Owner uses an instructional system design (ISD) process that takes into consideration the training audience, tasks, conditions, and standards needed to produce the requirements (e.g. Analysis, Design, Development, Implementation and Evaluation). An agile best practice is to express the requirements as *epics* and *user stories* (Beck, 1999). A user story is a description of a requirement from a user's point of view. A user story has the following structure:

As a <Role> I want <Requirement> so that <Reason / Return on Investment>

An epic is a large user story that cannot be delivered as defined within a single Sprint. Epics are typically decomposed into smaller user stories. There is no standard structure for epics, but many teams use the user story format. Once the Scrum Team has a well populated Product Backlog, the Backlog Refinement event typically takes place toward the middle of a Sprint.

A Sprint begins with the Sprint Planning event. The Scrum Team collaboratively plans the work the Development Team will perform at the Sprint Planning event. The input to this event is the *Sprint Goal*, the objective of the Sprint, and proposed *Sprint Backlog* which is the selection of requirements that would realize the Sprint Goal, if implemented. Before terrain development can begin, a terrain play-box and the desired training areas must be provided to the Scrum team prior to this event. This allows the Development Team to determine what terrain data is already available in their existing terrain data repository and what external GIS sources are available. During this event, the Product Owner discusses the Sprint Goal and the proposed Sprint Backlog. For terrain development, this would be a concept of operations (CONOPS) in the terrain play-box and user stories that describe how the training audience will interact with the environment. During this time, the Scrum Team has an opportunity to ask the Product Owner questions about the proposed Sprint Backlog. Once the Development Team understands the requirements, they choose the initial Sprint Backlog, decompose each requirement into technical tasks (e.g. create feature), and estimate the time required for each task. Task estimates should require one day or less to show steady progress at the Daily Scrum events. When the Development Team reaches their work limit and the Product Owner desires more features in the Product, the Development Team proposes tradeoffs to the Product Owner. The Development Team then finalizes the Sprint Backlog which is not changed during the Sprint. The output of this meeting is the final Sprint Backlog along with the work needed to deliver the capability.

Next, the Development Team begins developing the increment within the duration of the Sprint. The Development Team meets at the Daily Scrum to plan work for the next 24 hours. At the end of the Sprint, the Scrum Team attends the Sprint Review to demonstrate the increment to key stakeholders and adjust the Product Backlog if needed. This gives the Product Owner and key stakeholders an opportunity to provide feedback on the product increment and to add more requirements to the Product Backlog. For terrain development, this would likely include the review of the terrain content as a standalone review as well as a functional evaluation of the terrain within the context of the training system. The Sprint ends with the Sprint Retrospective which gives the Scrum Team the opportunity to reflect on what went well and what did not, and then to create a plan for improvements in the next Sprint.

The following summarizes the standard steps the Terrain Developers would execute for a Sprint.

1. Review Existing Terrain Data (Prior to Sprint Planning)
The Terrain Developer identifies what data is available in the terrain data repository for the desired terrain play-box. Performing this step ahead of time helps the Scrum team better understand what features already exist.

2. Discover External Sources (Prior to Sprint Planning)
The Terrain Developer identifies what external sources are available for the desired terrain play-box. If a feature is required and not available in the existing terrain data repository, the new source will be conflated with the existing source. Evaluating available sources based on geographic location is critical for long lead data purchases.

3. Derive Geospatial Requirements (Sprint Planning)
The Product Owner presents the user stories to the Development Team. The user stories describe how the training audience interacts with the environment. The Development Team derives the geospatial requirements from the user stories.

4. Suggest External Sources (Sprint Planning)
If new features are required, the Terrain Developer investigates sources where the required content may be obtained. This will include investigating government, commercial and open source data providers.

5. Provide Tradeoff to Collect or Generate (Sprint Planning)
Based on the inventory of existing terrain content and alternative sources identified to address any shortfalls, a make/buy trade is prepared. Figure 3 shows the analysis process flow used to determine our recommended approach.

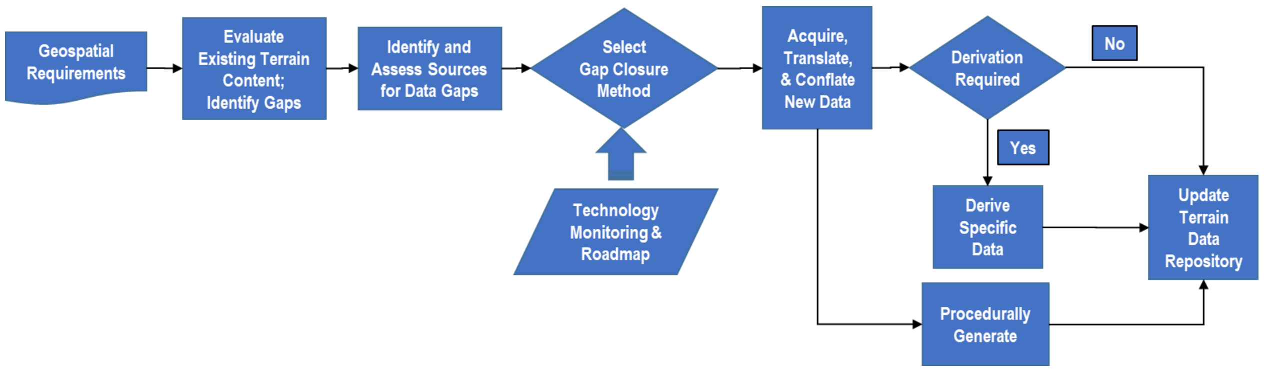
Geospatial Requirements
Evaluate Existing Terrain Content; Identify Gaps
Identify and Assess Sources for Data Gaps
Select Gap Closure Method
Technology Monitoring & Roadmap
Acquire, Translate, & Conflate New Data
Derivation Required
No
Yes
Derive Specific Data
Update Terrain Data Repository
Procedurally Generate

Figure 3 – New terrain data need analysis flow

Determining the best method to provide missing data involves evaluating a number of possible approaches. If the data is available via an open source, then translation and conflation will constitute the effort. If the data is available from a commercial source, the cost and time associated with obtaining the data must be considered in the trade. If the data is available from a government source, the time to obtain the approvals to use the data must be considered in the Sprint schedule. Another possibility is that the data is not directly available but can be derived through computation or fusing of multiple data sources. If so, the development of tools must be factored into the terrain generation process.

Most Terrain Developers monitor research and technology advancements to maintain technology roadmaps which are essential to planning efficiency improvements. When multiple candidate solutions exist, they analyze alternative trades, taking into consideration the cost of developing new tools and the overall cost of executing the generation process. Procedural generation of data is often a cost advantage, providing a plausible representation of the real world at a fraction of the cost to collect and generate extremely accurate representations. A blend of the two approaches may be required where high fidelity to the real world is provided only for limited, high-value training areas or key buildings within a megacity.

6. Create Features (Development)
If the desired content is not readily available, it must be created. If additional features are needed, vector editing will be employed to extract features from imagery, maps or photographs. If a new airport is required, it will be built from airport layout and lighting specifications. If new 3D models are required, the models will be built from drawings or photographs.

7. Merge with World DB (Development)
Once the features have been collected or generated, the next step is to merge the new features into the existing terrain repository. This will allow the new features to be available for future terrain databases.

8. Generate Runtime System Format (Development)
Once the new data is prepared and integrated into the existing terrain data repository, the Terrain Developer produces the required runtime databases using automated Database Generation System (DBGS) tools and processes. For most runtime formats this process is almost completely automated. Test tools are used to evaluate the readiness of the runtime content.

9. Demonstrate the Terrain Database (Sprint Review)
The Sprint concludes with a demonstration of the terrain database to the Product Owner and key stakeholders. This includes a standalone review of the terrain content and a functional evaluation of the terrain with the target training system. The Development Team will iterate over the core features in the database and give the stakeholders an opportunity to provide feedback and add new requirements to the Product Backlog.

10. Improve Our Terrain Production Process (Sprint Retrospective)
To ensure that the Terrain Developer provides the best possible product at the end of each Sprint, an evaluation of the Sprint results will be conducted. This will focus on an evaluation of the sources used, technologies employed, and the user determination of the results. The quality of the source data may be assessed or the fidelity of the creation process itself may be deconstructed. In the end, identifying areas of improvement is critical to providing the best results in the subsequent Sprint.

EXAMPLE SCENARIOS

In the following scenarios we show examples of how the Agile Terrain Development Process would be applied. As mentioned before, we assume the Product Owner has used an ISD process that considers the training audience, tasks, standards, and conditions when developing the user stories (requirements).

Scenario 1: Post-Hurricane Support

This scenario explores how the process is used to satisfy a system requirement for training first responders in a post-hurricane search and rescue operation. Table 1 identifies sample training tasks and conditions for Post-Hurricane Support training that was adapted from the National Planning Scenario Book. For this scenario, we assume the training will occur in a geographic area similar to New Orleans, Louisiana as shown in Figure 4.

Figure 4 Flooding in New Orleans, Louisiana (FM 3-06: Urban Operations, 2006)

Table 1 Post Hurricane Support training tasks and conditions

Task(s)	Condition
ART 7.4.3 – Provide Other Support as Required (air ambulance, search and rescue, safety and traffic control, temporary supplemental housing) (Army, ART 7.4.3 Provide Other Support As Required). ART 7.3.3.1 – Provide Essential Civil Services (providing food, water, shelter, and medical support until local civil services are restored (Army, ART 7.3.3.1 Provide Essential Civil Services).	Continental United States (CONUS) - Hurricane is over and rainfall has finished. Water has flooded low lying areas and stranded people throughout the area that require evacuation assistance. Significant areas are without power.

In applying the agile process to this scenario, the following steps would be performed:

1. **Review Existing Terrain Data** – Based on the defined terrain play-box, the existing terrain data repository would be searched for relevant data and provide a summary of the findings. For a flood scenario, the Terrain Developer reviews the terrain repository to ensure that the data include features such as roads, buildings, waterways, dams, and levees. Since this is a post hurricane search and rescue operation within CONUS, geocoded locations may be required. The Terrain Developer determines if this type of data is already available in the Terrain repository.

2. **Discover External Sources** – For this CONUS post hurricane scenario, the Terrain Developer likely looks to government organizations for data. A search of state, country, city, and ward related data is conducted to identify parcel maps, flood zone definitions, and detail elevation data defining low lying areas. The Terrain Developer likely will not look at commercial sources because there is generally high quality and plentiful geospatial data available within CONUS.

3. **Derive Geospatial Requirements** – Example epics and associated user stories for this scenario could include:

    Epic: As a helicopter pilot, I want to identify people to rescue from the air.
        User Story: As a helicopter pilot, I want to pick up a family from a damaged building.

    Epic: As a police officer, I need to perform door to door search.
        User Story: As a police officer, I need to clear buildings and mark them as clear.

    The user stories are derived from discussion during backlog refinement and Sprint Planning. For example, if a helicopter pilot is searching for people to rescue from the air, the pilot needs a way to find which locations need assistance. By having a building with a “Help Me” sign on the roof, the pilot could easily find these locations.

As ground crews, such as local police or the National Guard, search damaged areas, they can also mark cleared buildings with a large X so that others do not re-search these locations. The buildings should also have a state portraying temporary repairs. Table 2 lists some sample derived data needs for this Sprint. For this example, we limit discussion to a single derived geospatial requirement to provide representation for damaged, repaired, assistance needed, cleared, and destroyed buildings. Required supporting data elements include: GIS vector feature data for building footprints, building attribution (e.g. building functions), and actual 3D building models Figure 5 and Figure 6 are examples of building states that portray rescue markings.

Table 2 Post Hurricane Support Potential Data of Interest

Potential Data of Interest
• Distress Signal of various kinds
• Debris from hurricane damage
• Representation of impediments to air and ground navigation
• Tree canopy data, including downed trees and limbs
• Representation of road networks including road flooding
• Buildings with multi-states (e.g. damaged, repaired, cleared, and destroyed)
• Data supporting population pattern-of-life

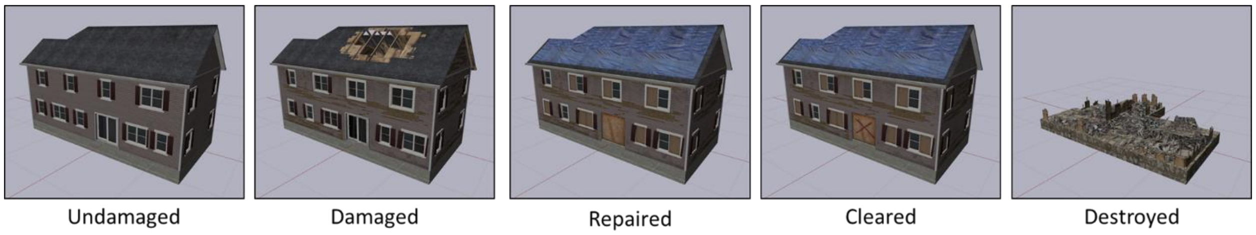

Figure 5 Example 3D Building Model Damage States

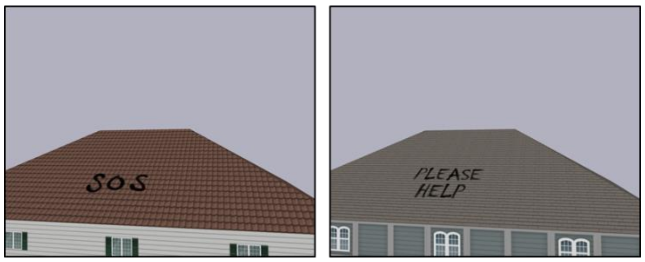

Figure 6 Example 3D Building Model Help Message

4. **Suggest External Sources** – For this example, we assume the existing terrain data repository does not contain building models with the required damage states. The identified data sources include 3D model repositories, both government owned and commercial. However, a source of the necessary building and house model states could not be identified to satisfy the requirements.

5. **Provide Tradeoff to Collect or Generate** – No models are available for purchase. In our example, the Terrain Developer has determined that multi-state models are required, and that they do not exist in the existing terrain repository. Also, they are not available off-the-shelf. The models can be created by the modeling team, or the Terrain Developer can use procedural model generation to provide the models. Procedural model generation is recommended.

6. **Create Features** – Procedural generation is the most cost-effective method to produce the large number of the 3D models required. To add the needed model states, the Development Team extends the existing

software functionality, adding incremental capabilities across multiple Sprints concluding with a customer demonstration at each increment. Leveraging procedural method makes follow-on terrain databases less expensive and able to be produced in shorter development time.

7. **Merge with World DB** - Once the multi-state 3D building models are created, they are integrated into the existing terrain repository by replacing the single-state models with the multi-state models. This includes adding any new construction rules and art assets to the repository.

8. **Generate Runtime System Format –** The Terrain Developer rapidly generates the required run-time databases using the identified DBGS and prepared build scripts. Additionally, the team uses quality and correlation test tools to verify the content is ready for user demonstration and evaluation.

9. **Demonstrate the Terrain Database –** Once the terrain is ready for the Sprint, it is presented to the Product Owner, first in a standalone review and then as part of a training scenario. Feedback from the Product Owner is documented and placed into the Backlog. In this example, the Product Owner noted that the 'rescue me' and 'SOS' markings were hard to identify at the desired distance and should be larger.

10. **Improve Terrain Production Process –** After the Product Owner review, the Terrain Developer conducts the process review. In this example, the Scrum Team recommends that in future Sprints the artwork should be made available to the Product Owner earlier in the process to minimize rework (e.g., the undersized 'rescue me' and 'SOS' markings).

Scenario 2: Bomb Detonation in Subway

This scenario explores how the process is used to satisfy a system requirement related to a bomb exploding in a city. In this example, we selected a bomb exploding in a subway system. Table 3 provides sample training tasks and conditions for the Bomb Detonation training that was adapted from the National Planning Scenario Book.

Table 3 Bomb Detonation training tasks and conditions

Task(s)	Condition
052-247-1228 - Perform a Rescue of an Injured or Entrapped Victim from a Collapsed Structure (052-247-1228 - Perform a Rescue of an Injured or Entrapped Victim from a Collapsed Structure, 2018).	CONUS - A bomb has been detonated in the subway of the financial district of a megacity mid-afternoon. Patrons are in panic. People are fleeing the immediate area using all forms of transportation available (taxi, subway, busses, walking). The subway is full of smoke.

For this scenario we would perform the following steps:

1. **Review Existing Terrain Data** – Based on the localized terrain in and surrounding the subway, the existing world terrain data repository is searched for relevant data and provide a summary of the findings. Given the subway as a key focus, it is important to find all available interior data including subway entrances, exits, and transportation lines. Example datasets of GIS data, 3D models, and previously developed terrain products are made available for viewing during Sprint Planning.

2. **Discover External Sources** – Identify any external sources available for the desired terrain data, such as subway system and transportation system.

3. **Derive Geospatial Requirements** – Based on training scenario discussion during Sprint Planning, the police or National Guard rescue victims in the subway, evacuate everyone from the subway area affected by bomb, and secure the subway entrance. They also need to search for secondary explosives. To provide a realistic backdrop, the simulation will model basic pattern of life flows. There are also some basic data needs including route lines, stop locations, and connection points. Example epics and associated user stories for this scenario could include:

Epic: As a Unit in the National Guard, I need to rescue victims in the subway.
User Story: As a National Guard Unit, I need to secure entrances so people can be evacuated.
User Story: As a National Guard Unit, I need to search for secondary explosives.

Based on these user stories, Table 4 lists some example derived data needs. This list would vary based on specific training requirements and training system capabilities.

Table 4 Bomb Detonation Sprint Potential Data of Interest

Potential Data of Interest
• Subsurface representation including subway entrances and subway tunnels.
• Representation of road networks including evacuation routes and speed limits
• Representation of the megacities mass transportation system, including evacuation routes, number of tracks/lines, train capacity
• Representation and attributes for surrounding building, including floorplan for training areas.
• Representation of the area hospitals or medical treatment areas.
• Data supporting infrastructure pattern-of-life

4. **Suggest External Sources** – The existing world dataset contains the basic footprint and identification of the subway and nearby hospital. Many sources for the layout of subways including stations and entrances are available.

5. **Provide Tradeoff to Collect or Generate** – Specific subway interior floorplans can be created by digitizing city subway system information. Data can be generated if not already collected.

6. **Create Features** – Subway transportation lines and surrounding roads are created by conflating the newest GIS vector features from key sources such as Open Street Map (OSM) and National Geospatial-Intelligence Agency (NGA). **Figure 7** shows example transportation route GIS vector feature data.

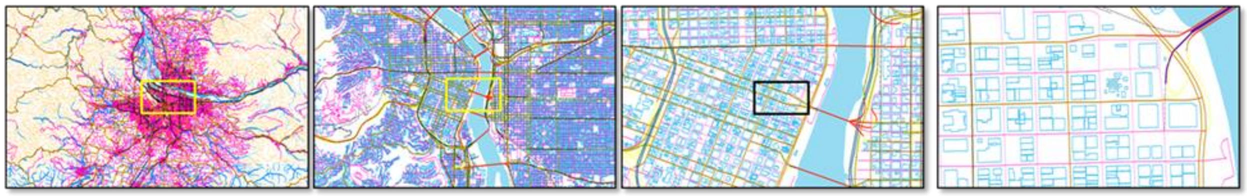

Figure 7 GIS Vector Feature Data Showing Transportation Infrastructure with Evaluation Routes

Missing transportation routes are extracted from the aerial imagery. The major 3D subway station structure model is likely manually developed. SE Core has developed a capability to generate the tunnel geometry and map visual textures for the subway tunnels, based on the vector features (see Figure 8).

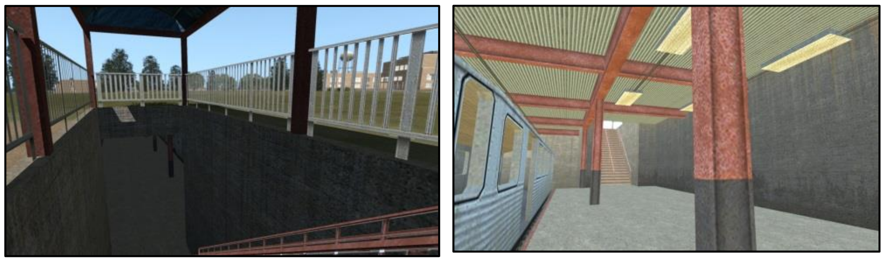

Figure 8 Examples of a Generated Subway Entrance and Station

7. **Merge with World DB –** Merge the subway features into the world database.

8. **Generate Runtime System Format –** Generate the runtime format by using the generate tools. This step includes testing the resulting terrain database to verify it satisfies the requirements modified or added in this Sprint.

9. **Demonstrate the Terrain Database (Sprint Review) –** Demonstrate the resulting terrain database to the customer on the training system. Use their feedback as inputs for the next Sprint. For this example, the customer requested building interiors to support training of evacuating surrounding builds. The buildings currently do not have interiors.

10. **Improve Our Terrain Production Process** – The Terrain Developers determine if there are any ways to improve the process from this Sprint. For example, creating standard textures for subways that could be reused in another training system that requires subways.

After this Sprint is completed, another one is started. In this example, the customer requested building interiors for evacuating the surrounding builds, since these building did not have interiors. This next Sprint follows the same steps with a focus on adding building interiors. Since there are few sources of 3D models with interiors, SE Core has developed tools that can procedurally generate geo-typical buildings with geo-typical interiors. If the training is general (not specific to a real building layout) then this approach works well. For the first two steps—Review Existing Terrain Data and Discover External Sources—models with interiors are sought. For the Derive Geospatial Requirements step, a new requirement of interior for selected buildings is added. For the fourth step, Suggest External Source, there are few 3D building models with interiors. For the next step, procedural generation is selected, based on existing tools being available. For the Create Features Step, 3D building models with interiors are procedurally generated. For the seventh step, these models are merged into the world database. Then, runtime formats are generated and tested. For Demonstrate the Terrain Database step, the new terrain database is run on the training system to show the insides of these new buildings. Ideas for improving process are then identified. Afterward, it is determined whether another Sprint is needed or the terrain database is ready for delivery.

CONCLUSION

Moving terrain database development and maintenance to an agile framework will greatly improve the timeliness and effectiveness of U.S. Army live, virtual and constructive training. Training no longer needs to be constrained to the functionality of a terrain database that was "built years ago." By leveraging an agile development process, terrain can be rapidly tailored to the meet an emerging training need. Providing a process to add or modify content of an existing terrain database is critical to support an alternate view of a specific location and satisfy an emerging training requirement. Terrain tailoring in minutes or hours, rather than days, weeks or months, is essential for supporting rapidly changing operational requirements.

The evolving training requirements associated with megacities and the costs associated with representing every feature in a single terrain database creates the need for an affordable process for adding specific content. The agile terrain database production process presented in this paper provides SE Core with a means to satisfy future U.S. Army's training needs.

REFERENCES

(2018). *052-247-1228 - Perform a Rescue of an Injured or Entrapped Victim from a Collapsed Structure.* US Army.

Army, U. (n.d.). *ART 7.3.3.1 Provide Essential Civil Services.* Retrieved from https://rdl.train.army.mil/catalog-ws/view/100.ATSC/3C6DFBEE-0422-4ECA-B6D9-4F7BED1AAAAC-1361574674053/report.pdf

Army, U. (n.d.). *ART 7.4.3 Provide Other Support As Required.* Retrieved from https://rdl.train.army.mil/catalog-ws/view/100.ATSC/EDF7D7C9-312C-412F-A9E8-5DC618862586-1361438772324/report.pdf

Beck, K. (1999). Embracing change with extreme programming. *Computer, 32*(10), 77.

Farrel, D., & Ward, M. (2016). *U.S. Army TRADOC G-2 Mad Scientist Megacities and Dense Urban Areas Initiative: Data Collection and Analysis.* Hampton: The MITRE Corporation.

(2006). *FM 3-06: Urban Operations.* US Army.

Howe, D. (2004). *PLANNING SCENARIOS: Executive Summaries.* The Homeland Security Council .

(2013). *Joint Publication 3-06: Joint Urban Operations.* Joint Staff.

(2005).*Katrina NOLA levee break FEMA.* Federal Emergency Management Agency. Retrieved June 14, 2018, from https://commons.wikimedia.org/wiki/File:Katrina_NOLA_levee_break_FEMA.jpg

Schwaber, K., & Sutherland, J. (2017). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game.

Sutherland, J. V., & Schwaber, K. (1995). Business object design and implementation: OOPSLA '95 workshop proceedings. *OOPSLA.* Michigan.