Ⅰ．未来を見据えて

STEM教育とは何か

子供にとって：

答えが複数ある問題に対して，STEM分野の知識や見方・考え方を統合的に働かせて，文脈に適した解決策をつくり・調査し・試行し・改善し・提案し・議論する一連の活動を通じて，それぞれの分野で期待される資質能力の育成のみならず，単独の分野の学習ではなし得なかった，全体性や相互関連性についての理解を深め，そうした理解に基づいた態度・学びに向かう人間性などを養う．

Society5.0

サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立する、人間中心の社会（内閣府）．「個別に公正最適化された学習」が成立すると同時に，時短できるため，STEAMのような探究型の学習に時間が割けるようになる（経産省）とされている．

AIは教師の「指導力」を学習することになるが，探究の指導はできるのか？

プログラミングとの関係性

STEMと共存するプログラミングはエンジニアリングか？，それともテクノロジーか？

テクノロジーの場合：プログラミング自体を子供がエンジニアリングするようになる．現時点では考えにくいが，プログラム自体の創造・試行・改善を将来的には行うことになる？

エンジニアリングの場合：目的を達成するためのプログラミング．例えば，ロボットを動かす，スマートハウスなど．ただし，現時点では多様な問題，多様な解決策に対応できていない．

何がこれまでと違うのか

- 教科の範囲を超えているが，従来の総合的な学習な時間よりは具体的に文脈を設定できる．日本は，既に技術の時間，総合の時間があるので，その時間でやることも考えられる．
- エンジニアリングに重きを置いていること．ものづくりとの違いは，「もの（T）」が一つではなく，多様性を認めていること，創造性につながること，「もの」は物だけでなく，アイデア，プロセス，システムを含んでいること．
- 回路，選挙，会社，音楽会，絵画，ダンスもT

個別最適化された学びとは

- 子供たちの得意・苦手に合わせて，学習内容を最適化すること．目標とする学習内容を達成するために，最適なドリルを行う．

かのように，考えられていますが…

- 子供たちの追究する内容に合わせて，提供する知識が最適化されるようになるべき．
- AIはあくまでも道具なので，それに人間が併せていく必要はない（プログラミングしかり）．
- むしろ，思考を発散する道具になるべき（IA）．

エンジニアリングとプログラミング

プログラミングとエンジニアリングそれぞれについて，十分検討が必要．それぞれの良さを打ち消し合わない統合の仕方ができるとよい．

- プログラミングが現時点で相性が良いのは算数
- 多様な問題，多様な解決策に対応するためには，思考の発散に対応しうるデバイスの準備
- 「”子供向け”にしすぎないシステム」の構築・利用と，それを「子供向け」に教えることの意義と方法の開発・発展．

何が期待されるか

- 学習への（関係者）全員参加
- 学びの意欲の向上（到達度は結果的に向上）
- 教師同士の共通の課題設定，教育課程全体を通じた背骨となる，学校としてのめあての達成．
- グランドデザインを地で行く授業づくり．
- 学校というテクノロジーが，世田谷区の教育のニーズに応えていくこと（エンジニアリング）．
- 学校，区の教育活動への保護者の関心・理解の向上，責任の共有の再確認．

STEMと最適化の関係

- EdTechで知識を学び，STEAMで活用するというように，固定しすぎないほうがよい．
- 個人の学習に最適化をするのがむしろSTEAM（スクール）におけるエンジニアリングの役割．
- 例えばSTEAMとしては，同じ教材(T)に取り組んでいても，それぞれの子で，学ぶ深さが出てくる．違学年が，同教材に取り組むことができる．
- STEAMスクールとしては，STEM教材が区のニーズに応えていることが学校を最適なものとしていく．ひいては，個人の学びにもつながる．

ヒューリスティックを用いる力

- 課題の発見，課題の追究，課題の解決といった探究の過程を振返りながら，探究を自らが進めている！という感覚を持つために，ヒューリスティック（ゆるめの方針）を自覚させる．
- こうじゃなきゃいけないというものではない．むしろ固定しすぎると創造性にはつながらない．行きつ戻りつ，上手く行かなかったら，変えていく（ようになる）のが理想．
- 最終的にできるのは中学校以上で良いが，芽を生やしたい←ここに生きるプログラミング教育

2．今日からできること，考えておくべきこと

STEM教育とは何か

- 教師にとって：研修機会；自分の得意とする分野から少し飛び出して，他の分野の内容・方法・見方・考え方を学ぶ機会．「個に応じた指導・学習・評価」を地で行く方略．教材づくりが，研修そのもの．
- 学校にとって：カリキュラム・マネジメント＆教員協働の場作り，一人ではできなかったことを複数で実現する．
- 区にとって：地域に開かれた学校づくり，区民が参加し，社会的に堅牢な知が生産される場所．

Society5.0

- 探究を指導する「コーチング」を毎日の授業から進め，練習しておく．
- コーチングの基本は，問いかけ（発問）．
- 発問が学習を規定するのではなく，子どもたちの疑問を導くものでありたい．
- 「（子供たちの）疑問の焦点」づくりを磨く．
- 発問のバリエーションを持つ．
- オープンな発問に対する子供の反応に慣れる．

プログラミングとの関係性

- アルゴリズム的思考は，後に課題を解決する「過程を管理する力」を養うものと捉える．
- プログラミングの活動とSTEAMの活動を混同・混合しすぎないでよい（それぞれの価値構築をまず優先する）．
- 手順を知ると，自由を持てる．変えてみたくなる．→どちらにも通じる．
- 教師の側がヒューリスティックを持つこと．
- 繰り返しになりますが，その順でなくてもよい．

これまでより

- 教師の役割は，知識の伝達者から，子供たちの学習を促進するファシリテーター，更には「世田谷区立〇〇〇学校」において通用する知の生産者に，子供と協同でなる．
- 失敗を許される学習環境づくり，教師の発言，クラスメイトの受容，保護者の理解．
- 教育の民主化：学びは学習者・保護者・教師がつくるものによりなっていき，学校・区・都・国はそれを支援する．

個別最適化された学びを受けて

- それぞれの子供にとって，最適な「問い（めあて）」は何か．具体的であればあるほど，個の学習は「決定」されてしまう．大きすぎると，具体的な指導の計画が難しくなる．→実は，一人ひとり最適な問いは異なる．
- 多様で非均質な子供たちの学習を含み込めるだけのめあての大きさと，そこまで個を支援できる具体的な手立てを持っておく必要がある．
- 知識の提供は，EdTechがやってくれるとすれば，個の支援に必要なことは何？

エンジニアリングとプログラミング

- エンジニアリングは「たまたま」できること，プログラミングは，「決めうちする」ことに意味がある．
- 違う意味で「ねばりづよさ」が求められる．
- 学習機会が「繰り返す」ことに，価値がある．
- 同じ教材で（ではなくても），同じことを繰り返しやって上達しているという感覚を創りたい．一度や二度では上手く行かないということでもある．→「失敗を許される」

何が期待されるか

- 学習への（関係者）全員参加
- ただしやっていることは，みんな違うという意識：No Child Left Behind（一人[のニーズ]も取り残さない）
- それでも，みんなで取り組める大きなめあては何か（多様性を含み込めるだけの大きさで，かつ取り組む姿が見える具体的なもの）
- 各学区の棚卸し．一体どんな身近なテクノロジーが教材となり得るか→各教科目線で

STEMと最適化の関係

- 同じ教材（テクノロジー）でも，どんなニーズに答えるために出来ているのかを多様に考える．→個の学びの多様なきっかけづくり
- “まなび”と”つくり”のバランス，これも個によって違うでしょう．
- できれば，どの子にもどちらも経験させたい．
- が，そもそもどちらが得意なのかも違っている．
- 得意な方から始めましょう．苦手な方は，後からついてくる．ぐらいのつもりで．自信優先．

ヒューリスティックを用いる力

- めあてと手順は，大海で失敗した時に戻ってくる港になる．
- めあて・手順を考える機会はいつだろう？
  →プログラミング教育！
- 理科の教科書では，方法（手順）が決まっている．→それでも考える時間を取ってみる．
- 多分，最初は上手く行かない
- →指導方法を考えてみる，改善してみる

＝エンジニアリングしてみる→研修につながる！